JOINT POLAR SATELLITE SYSTEM (JPSS) COMMON GROUND SYSTEM (CGS)

IDPS ADL SOFTWARE USER'S MANUAL PART 1

SOFTWARE RUNTIME ENVIRONMENT AND INSTALLATION

CDRL No. A032

RAYTHEON COMPANY
INTELLIGENCE AND INFORMATION SYSTEMS (IIS)
JPSS PROGRAM
AURORA, COLORADO

"Raytheon Proprietary
© 2011 - 2012 Raytheon Company
Sponsored by the United States Government
Under Contract No. NNG10XA03C

Notice to Government Users:
Refer to FAR 52.227-14, "RIGHTS IN DATA — GENERAL" (Dec. 2007), as modified by
NASA FAR Supplement 1852.227-14 – Alternate II (Dec. 2007)
and Alternate III (Dec. 2007), paragraph (c)(1)(iii),
for Government's license rights in this data or software.

Notice to Non-Government Users:
Subject to Proprietary Information Agreement or Other Non-Disclosure Agreement"

TITLE: JOINT POLAR SATELLITE SYSTEM (JPSS) COMMON GROUND SYSTEM (CGS) IDPS ADL SOFTWARE USER'S MANUAL PART 1 SOFTWARE RUNTIME ENVIRONMENT AND INSTALLATION

See pdf file for digital signatures

APPROVAL SIGNATURES:
(Signatures indicate compliance with these requirements)

_____

William J Sullivan Date
Program Manager

_____

Frank Zorn Date
Sustainment IPT Lead

_____

Harvey Y. Lee Date
IDPS Sustainment Lead

_____

Justin E Lee Date
Software Manager

_____

Sheryl Fertman Date
IIS Quality

Revision/Change History

Revision	Document Date	Revision/Change Description	Pages Affected
---	30 Nov 2011	Released per ECR-IDP-004060	All
A (Draft)	27 Apr 2012	Updated for ADL Phase 4.0 per ECR-IDP-004680 ADL 4.0: ADL User Manual	All

Table of Contents

1 SCOPE 1
1.1 Identification 1
1.2 System Overview 1
1.3 Document Overview 6
1.4 Targeted Platform 6
1.5 Workstation / Server Intended Use 7
2 REFERENCE DOCUMENTS 8
3 SOFTWARE SUMMARY 9
3.1 Software Inventory 9
3.1.1 Libraries 9
3.1.2 Executables 15
3.1.3 Guide List Files 17
3.2 Software Environment 18
3.2.1 COTS Products 18
3.2.2 Environment Variables 20
3.2.3 Data Sources 21
3.2.4 Configuration Files 21
3.2.5 Batch Files / Command Line Parameters 30
3.2.6 Drives 30
3.2.7 User/Service Accounts 30
4 INSTALLATION AND SETUP 31
4.1 Prerequisites 31
4.2 Installing the Software 31
4.3 Configuring the Software 37
5 MISCELLANEOUS NOTES 38
5.1 Possible Compiler Warnings 38

List of Figures

Figure 1.2-1 ADL With Algorithm Packages 3
Figure 1.2-2 ADL Architecture versus IDPS Operational Architecture 5
Figure 1.2-3 ADL Framework versus IDPS Operational framework 6

List of Tables

Table 2-1 Referenced Documents 8
Table 3.1.1-1 Software Inventory – Runtime Libraries 9
Table 3.1.2-1 Software Inventory – Algorithm Executables 15
Table 3.1.2-2 Software Inventory – Tool Executables 16
Table 3.1.3-1 Guide List Files 17
Table 3.2-1 Compatible OS and Compilers 18
Table 3.2.1-1 ADL COTS Software 18
Table 3.2.1-2 IDPS Algorithm COTS Software 19
Table 3.2.2-1 Environment Variables 20
Table 3.2.4-1 Configuration Files 22
Table 3.2.4-2 Algorithm Execution Configurable Items 29

1 SCOPE

1.1 Identification

The Algorithm Development Library (ADL) Software User's Manual Part I describes the architecture, installation and setup of the Algorithm Development Library. This document has been prepared in support of the ADL Software, of the Joint Polar Satellite System (JPSS) and the NPOESS Preparatory Project (NPP).

1.2 System Overview

The Joint Polar Satellite System (JPSS) is a System-of-Systems architecture whose mission is to provide military and civilian agencies with environmental, meteorological, and climatological data and products. The Common Ground System (CGS) provides command and control of the National Polar-orbiting Operational Environmental Satellite System (NPOESS) Preparatory Project (NPP) and JPSS spacecraft and instruments. The CGS transports commands, telemetry, and mission data between the space, ground control, and processing facilities and the Customer locations, and enables mission planning, asset management, and generation and delivery of the mission products.

The CGS Program is responsible for the design, implementation, integration, test, deployment, and transition of the three JPSS Ground Segments: the Command, Control, and Communications Segment (C3S), the Interface Data Processing Segment (IDPS), and the Field Terminal Segment (FTS). C3S manages the overall JPSS CGS mission, including active management and accounting of mission data, from storage on the satellite solid-state recorder to delivery to the IDPS. The IDPS is responsible for receiving raw mission data from C3S and for creating and delivering useable environmental products to Users. FTS provides hardware and software specifications needed to build JPSS CGS-compliant field terminals (FTs), along with updated processing software designed to handle direct broadcast data streams.

There is an Interface Data Processor (IDP) deployed at each Operational Central, located at National Environmental Satellite, Data, and Information Service (NESDIS), Air Force Weather Agency (AFWA), Fleet Numerical Meteorology and Oceanography Center (FNMOC), and Naval Oceanographic Office (NAVO). The primary function of each IDP is to ingest and process Stored Mission Data (SMD) into data products and then to deliver the data products to internal and external users and systems. The IDP provides the data products to the Centrals based on system level latency requirements. Additionally, the IDP supports long term monitoring, calibration and validation of the products and satellite sensors.

The IDP ingests and processes SMD received from the Command, Control, and Communications Segment (C3S). The artifacts from satellite on-board storage and ground communication routing are removed and the resulting Application Packets (APs) are then stored in Raw Data Records (RDRs) as raw data on a per sensor and spacecraft basis. Each IDP also ingests Auxiliary and Ancillary data needed in the processing of the SMD. The RDRs, Auxiliary data, and Ancillary data are subsequently processed to create Sensor Data Records

(SDRs), Intermediate Products (IPs), Application Related Products (ARPs), and Environmental Data Records (EDRs). The created data products, excluding RDRs, are normally geolocated with the geospatial information. Once the data records are created, they are delivered, along with the a subset of the Auxiliary and Ancillary data, as requested to the users, including the Centrals, the NOAA Science Investigator-led Processing System (NSIPS), the C3S Payload Support Tool (PST), the NASA Science Data Segment (SDS), and the NOAA Comprehensive Large Array-data Stewardship System (CLASS).

The Algorithm Development Library provides both a core algorithm framework in which to develop and execute new algorithms or existing ADL compatible algorithms, and an optional IDPS algorithm package. ADL compatible algorithms, in this document, refer to any algorithm that can run within the core algorithm framework, even if the algorithm is developed outside of the Common Ground System (CGS). When used to develop new algorithms, the ADL core algorithm framework, in general, absolves the user from writing common code modules, such as modules to read and write inputs and outputs to disk, as long as the user defines the format for each of the algorithm's inputs and outputs in an ADL-defined XML format. The IDPS algorithm package includes the IDPS VIIRS, CrIS, ATMS, and OMPS algorithm software. Please note that the ADL IDPS algorithm package does not operate as a standalone package; It must be used within the ADL core algorithm framework.

While the ADL core algorithm framework can be used to develop algorithms outside of the Common Ground System (CGS), its main purpose is to aid in the development of algorithms destined to be used in the CGS IDPS. This is because the ADL core framework provides the same interfaces as the CGS IDPS, allowing the algorithms developed using the ADL core framework to be easily "plugged" into the CGS IDPS.

The current release of ADL contains several key components as depicted by Figure 1.2-1:

- Core algorithm framework
- Generic granulation component
- Common geolocation component
- Support for processing algorithm diagnostic inputs and outputs (Please note that unless otherwise specified, the terms "input(s)" and "output(s)" include diagnostic input(s) and output(s), respectively, throughout the remainder of this document.)
- Support for diagnostic code blocks
- Conversion of native JPSS Mission Support data (MSD) to IDPS internal file formats
- Ability to create and extract IDPS compliant HDF products
- An XML Editor tool for defining new or modifying existing XML product format definitions
- A binary Editor tool for modifying binary data products.

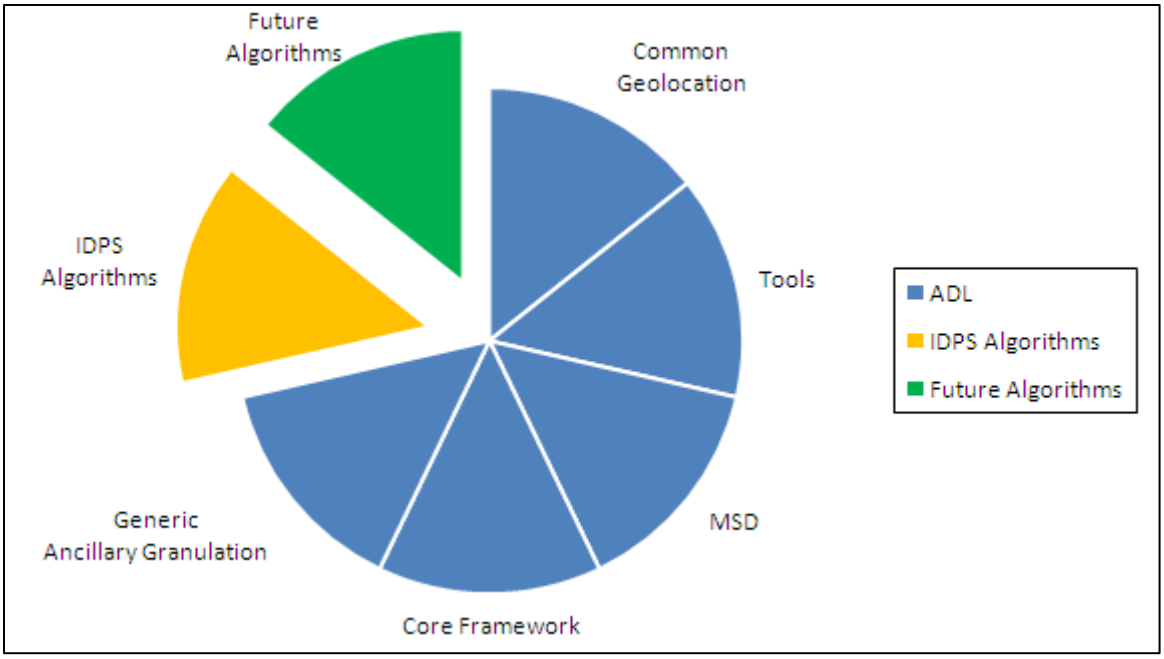


Figure 1.2-1 ADL With Algorithm Packages

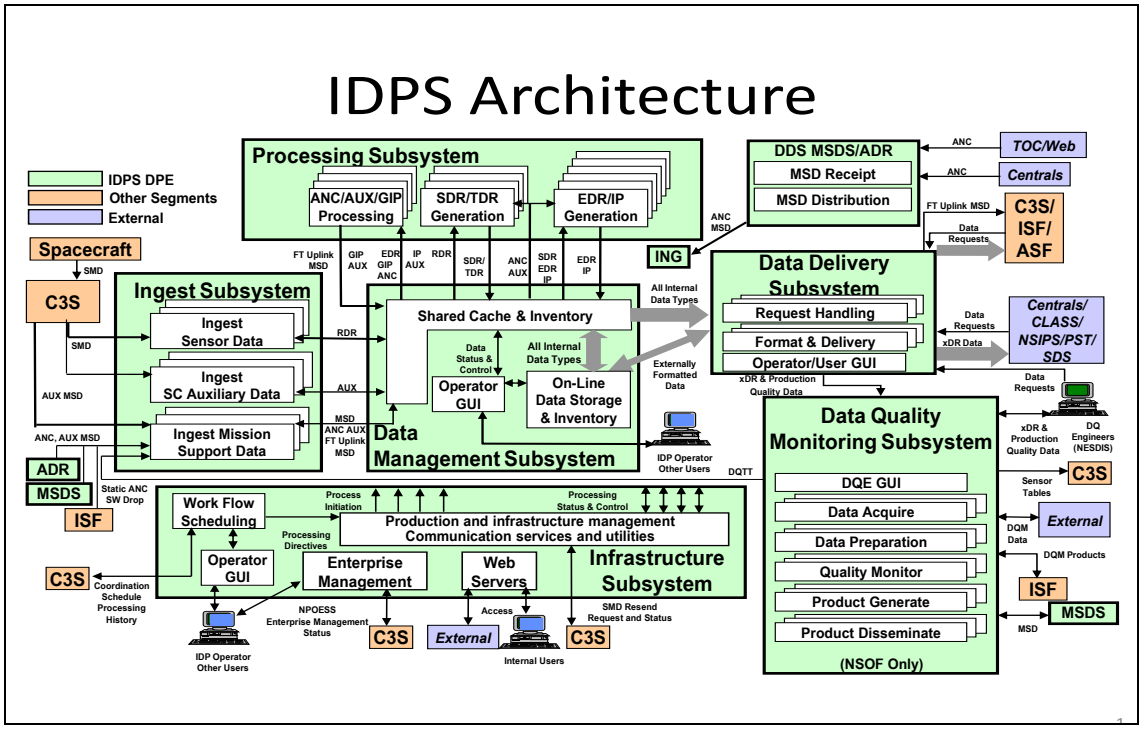
The ADL core algorithm framework can be used by scientific algorithm code developers (hereafter called “algorithm developers”) to help integrate Government Furnished Property (GFP) algorithm code more rapidly into the JPSS Interface Data Processing Segment (IDPS) as part of the Processing Subsystem (PRO). This allows for improved efficiencies when integrating calibration and validation algorithm changes into the operational IDPS for those operational algorithms which are ADL compatible.

As part of the science-to-operations code conversion (sci2ops) process, IDPS engineers have adapted science algorithms into an Input (“I”)-Processing (“P”)-Output (“O”) model. For algorithm code under development, the ADL core algorithm framework mimics the interfaces of the “I” and “O” portions of the IDPS PRO Common software to other IDPS SIs. This is accomplished by providing rewritten DMS software that accomplishes the same “I” and “O” tasks that the current IDPS DMS software provides but without the interaction with an Oracle database and the Infrastructure Subsystem (INF),the operational subsystem responsible for behind the scenes infrastructure. The functionality to the algorithm is transparent, although the implementation in some aspects will be different, because the ADL core algorithm framework uses the IDPS Data Management Subsystem (DMS) APIs with modifications to the implementation to allow for file based input and output. The use of the ADL core algorithm framework as a development environment for GFP algorithm code will result in a decrease in time spent by IDPS for sci2ops conversion. The use of the ADL core algorithm framework will promote a greater collaboration between the algorithm developers, IDPS development engineers, and the operational users.

The ADL environment, with the IDPS algorithm package, is used to develop and debug the algorithm code items on computing platforms outside of IDPS. Currently supported platforms are AIX running on POWER (big endian) hardware and Red Hat Enterprise Linux 5 running on x86-64 (little endian) hardware. Supported compilers are either xlC/xlf or GNU GCC compilers.

When linked to the ADL core algorithm framework, the algorithm code item is able to execute as if it were inside the IDPS without invoking additional proprietary software. Given the same inputs, the algorithm code item in the ADL environment produces an equivalent result to that of the algorithm code item running in the IDPS environment. In this context, "equivalent result" means equal except for allowable differences due to round off error, different compilers, rehosting, etc. Moreover, given the same inputs, an operational IDPS algorithm code item executed by the IDPS and the same algorithm code item running in the ADL environment on a computational platform outside of the IDPS, produce equivalent results. ADL core algorithm framework is intended for serial processing of algorithm code items (no parallel processing or threading).

The figure below provides a context diagram for the ADL framework and depicts its similarities to the IDPS architecture.



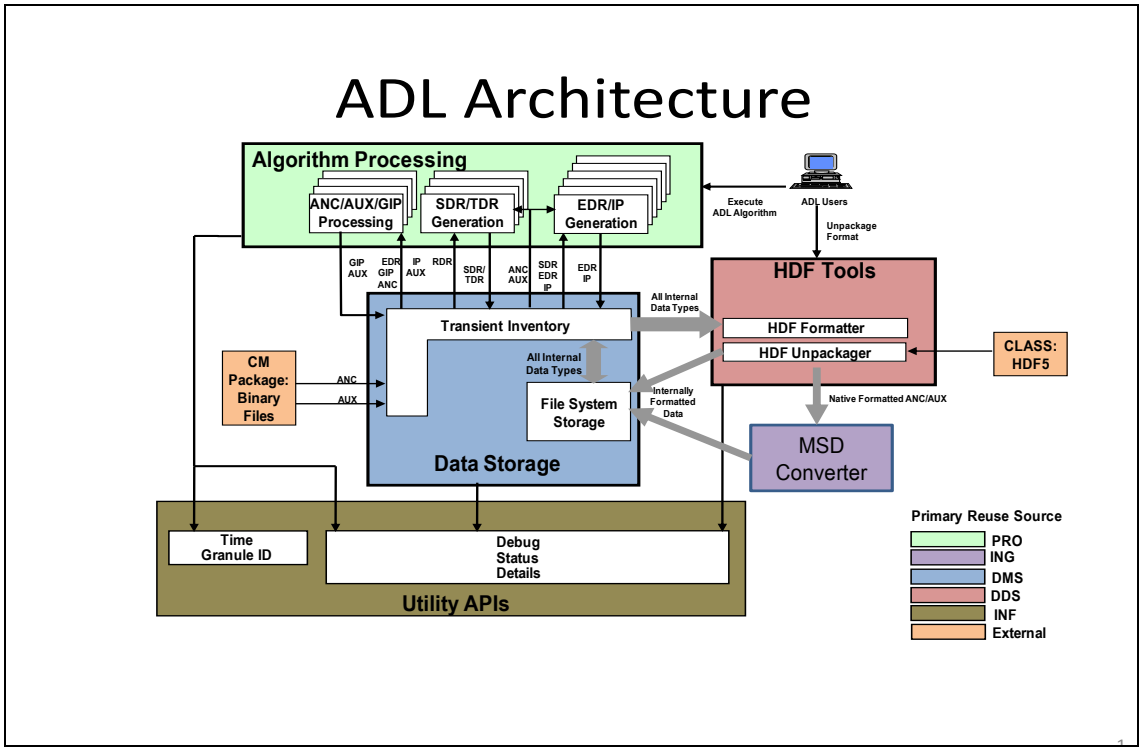


Figure 1.2-2 ADL Architecture versus IDPS Operational Architecture

The figure below portrays how the “P” or algorithm processing software can be plugged into either the ADL or IDPS frameworks, without requiring additional modifications. The “P” block here represents the same algorithm software in both cases.

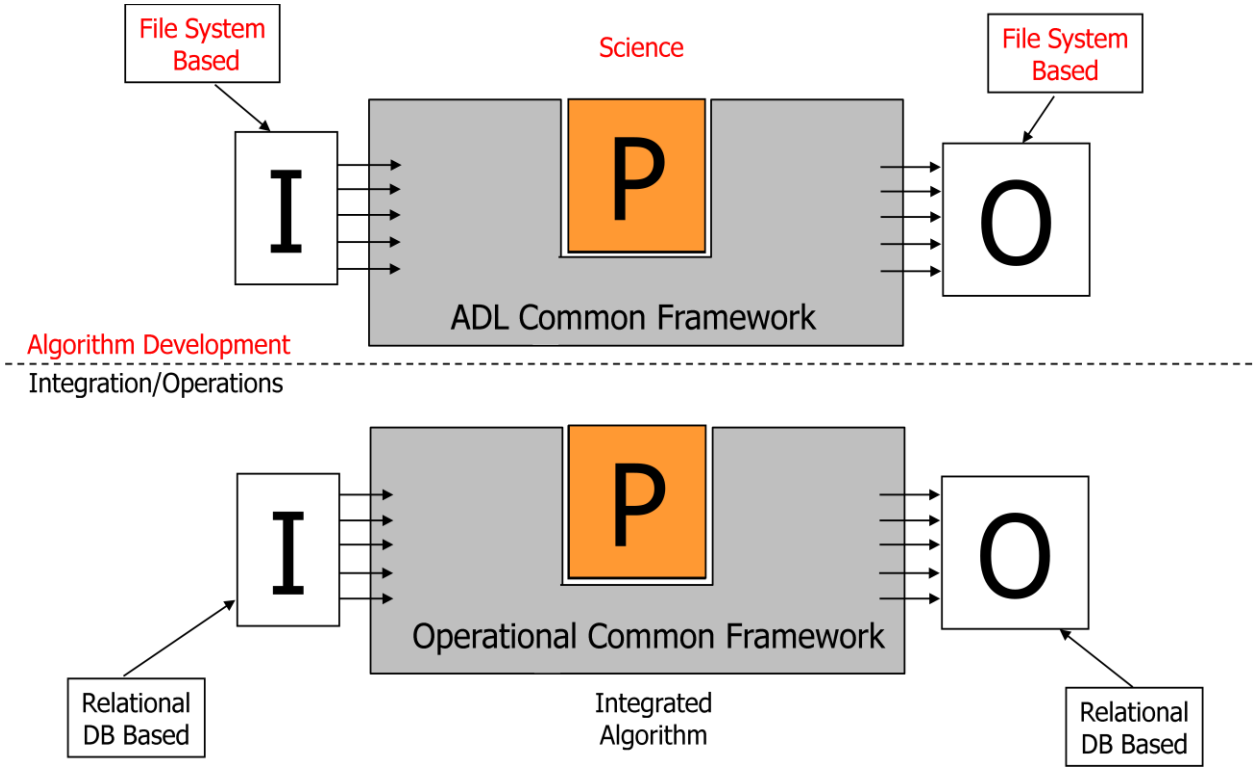


Figure 1.2-3 ADL Framework versus IDPS Operational framework

1.3 Document Overview

This document describes the architecture, installation, and setup of the ADL software.

Section 1 – Basic overview of ADL
Section 2 – References used in this document
Section 3 – Software Summary
    Section 3.1-- Software Inventory
    Section 3.2– Software Environment
Section 4 – Installation and setup including installation and configuration
Section 5 – Miscellaneous Notes

1.4 Targeted Platform

The ADL software is targeted to run on RHEL/x86-64 (little endian) and AIX/POWER (big endian) platforms.

1.5 Workstation / Server Intended Use

All ADL software installs from and runs and executes using the operating system command line prompt.

2 REFERENCE DOCUMENTS

This section identifies the documents referenced in this document.

Table 2-1 Referenced Documents

Document #	Document Title
UG60917-IDP-034	IDPS ADL Software User's Manual Part 2
UG60917-IDP-027	IDPS Processing Software Configuration Guide User Manual
UG60917-IDP-036	PRO XML Editor Users Manual
UG60917-IDP-042	IDPS PRO Binary Editor Tool User's Manual
UG60917-IDP-030	IDPS ING Software User's Manual Part 2
LI60917-GND-005	JPSS CGS Acronyms & Glossary

3 SOFTWARE SUMMARY

3.1 Software Inventory

3.1.1 Libraries

ADL requires numerous libraries in order to execute a new algorithm or the ADL compatible IDPS algorithms. The following table lists ADL libraries that are compiled in the ADL development environment. These are shared and static libraries that are compiled in $ADL_HOME/lib at build time and then linked into the ADL executables.

Table 3.1.1-1 Software Inventory – Runtime Libraries

Library Name	Description
libADLHDF.so, libADLHDF.a	ADL HDF library
libADLPacker.so, libADLPacker.a	ADL HDF Packer library
libADLUnpacker.so, libADLUnpacker.a	ADL HDF Unpacker library
libADLUtil.so, libADLUtil.a	ADL Utility library
libAdlCmnINF.so, libAdlCmnINF.a	ADL Common INF library
libAdlEdrAlgorithmTemplate.so, libAdlEdrAlgorithmTemplate.a	EDR Algorithm Template library
libAdlEdrAlgorithmTemplate2.so, libAdlEdrAlgorithmTemplate2.a	EDR Algorithm Template 2 library
libAdlScienceAppl.a, libAdlScienceAppl.so	Science Application library
libAdlScienceDictEntries.a, libAdlScienceDictEntries.so	Science Dictionary Entries library
libAdlSdrAlgorithmTemplate.so, libAdlSdrAlgorithmTemplate.a	SDR Algorithm Template library
libAdlSdrCmnGeoFInterfaces.so, libAdlSdrCmnGeoFInterfaces.a	SDR Common Geolocation Fortran Interface library
libAdlSrcAutogen.a	ADL Source Generation library
libAdlTemplateAppl.a, libAdlTemplateAppl.so	Template Application library
libAdlTemplateDictEntries.a, libAdlTemplateDictEntries.so	Template Dictionary Entries library
libDmApi.so, libDmApi.a	DMS API Library
libDmCoreDb.a, libDmCoreDb.so	DMS Core Database library
libDmCoreLibrary.so, libDmCoreLibrary.a	DMS Core library
libDmImInventory.so, libDmImInventory.a	DMS Inventory Management library
libDmImInventoryAdaptation.so, libDmImInventoryAdaptation.a	DMS Inventory Management Adaptation library
libDmMgmt.so, libDmMgmt.a	DMS Management library
libDmSmStorage.so, libDmSmStorage.a	DMS Storage Management library
libDmSmStorageAdaptation.so, libDmSmStorageAdaptation.a	DMS Storage Management Adaptation library
libInfCmnExc.so, libInfCmnExc.a	INF Common Exception library
libInfCmnUtil.so, libInfCmnUtil.a	INF Common Utility library
libInfTk.a, libInfTk.so	INF TK library
libInfUtilGran.a, libInfUtilGran.so	INF Granule Utility library
libInfUtil_Cfg.so, libInfUtil_Cfg.a	INF Configuration Utility library
libInfUtil_Checksum.so, libInfUtil_Checksum.a	INF Checksum Utility library

Library Name	Description
libInfUtil_Dbg.so, libInfUtil_Dbg.a	INF Debug Utility library
libInfUtil_Perf.a, libInfUtil_Perf.so	INF Performance Utility library
libInfUtil_Tim.so, libInfUtil_Tim.a	INF Time Utility library
libIngMsdCoefficients.so, libIngMsdCoefficients.a	ING MSD Coefficients library
libIngMsdCommon_Abstract.so, libIngMsdCommon_Abstract.a	ING MSD Abstract library
libIngMsdCommon_Exceptions.so, libIngMsdCommon_Exceptions.a	ING MSD Common Exceptions library
libIngMsdCommon_GribConv.so, libIngMsdCommon_GribConv.a	ING MSD Common Grib Converter library
libIngMsdCommon_Main.so, libIngMsdCommon_Main.a	ING MSD Common Main library
libIngMsdCommon_XmlConv.so, libIngMsdCommon_XmlConv.a	ING MSD Common XML Converter library
libIngMsdEarthOrientation.so, libIngMsdEarthOrientation.a	ING MSD Earth Orientation library
libIngMsdLUT.so, libIngMsdLUT.a	ING MSD LUT library
libIngMsdMissionSchedule.so, libIngMsdMissionSchedule.a	ING MSD Mission Schedule library
libIngMsdNAAPS.so, libIngMsdNAAPS.a	ING MSD NAAPS library
libIngMsdNCEP.so, libIngMsdNCEP.a	ING MSD NCEP library
libIngMsdNOGAPS.so, libIngMsdNOGAPS.a	ING MSD NOGAPS library
libIngMsdRevolutionTable.so, libIngMsdRevolutionTable.a	ING MSD Revolution Table library
libIngMsdSCConfigurationDB.so, libIngMsdSCConfigurationDB.a	ING MSD SC Configuration library
libIngMsdSeededIP.so, libIngMsdSeededIP.a	ING MSD Seeded IP library
libIngMsdStaticAnc.so, libIngMsdStaticAnc.a	ING MSD Static ANC library
libIngMsdThresholds.so, libIngMsdThreshold.a	ING MSD Thresholds library
libIngMsdTLE.so, libIngMsdTLE.a	ING MSD TLE library
libNhf_Rtm_Lut.a	*Net Heat Flux Radiative Transfer Model Look Up Table library
libNhf_coare_ice.a	*Net Heat Flux Coupled Ocean-Atmosphere Response Experiment Ice library
libNhf_coare_ice_mod.a	*Net Heat Flux Coupled Ocean-Atmosphere Response Experiment Ice Module library
libNhf_coare_water.a	*Net Heat Flux Coupled Ocean-Atmosphere Response Experiment Water library
libNhf_coare_water_mod.a	*Net Heat Flux Coupled Ocean-Atmosphere Response Experiment Water Module library
libNhf_coart_24band.a	*Net Heat Flux Coupled Ocean and Atmosphere Radiative Transfer library
libNhf_coart_24band_mod.a	*Net Heat Flux Coupled Ocean and Atmosphere Radiative Transfer Module library
libNhf_rrtmg_lw.a	*Net Heat Flux Rapid Radiative Transfer Model Long Wave library
libNhf_rrtmg_mods.a	*Net Heat Flux Rapid Radiative Transfer Model Module library
libNhf_rrtmg_sw.a	*Net Heat Flux Rapid Radiative Transfer Model Short Wave library
libProAncCrisController.a, libProAncCrisController.so	*CrIS Ancillary Controller library

Library Name	Description
libProAncGenericGranAnc.a, libProAncGenericGranAnc.so	Generic Granulate Ancillary library
libProAncGranulateCris.a, libProAncGranulateCris.so	*Granulate Ancillary CrIS library
libProAncGranulateOmpsTc.a, libProAncGranulateOmpsTc.so	*Granulate Ancillary OMPS TC library
libProAncGranulateViirs.a, libProAncGranulateViirs.so	*Granulate Ancillary VIIRS library
libProAncGranulateViirsUtil.a, libProAncGranulateViirsUtil.so	*Granulate Ancillary VIIRS Utility library
libProAncOmpsNpController.a, libProAncOmpsNpController.so	*OMPS NP Ancillary Controller library
libProAncOmpsTcController.a, libProAncOmpsTcController.so	*OMPS TC Ancillary Controller library
libProAncViirsController.a, libProAncViirsController.so	*VIIRS Ancillary Controller library
libProCmnAdj.a, libProCmnAdj.so	PRO Common Adjacency library
libProCmnAncAppl.a, libProCmnAncAppl.so	*ANC Application library
libProCmnAncDictEntries.a, libProCmnAncDictEntries.so	ANC Dictionary Entries library
libProCmnAtmsAppl.so, libProCmnAtmsAppl.a	*ATMS Application library
libProCmnCrimssAppl.so, libProCmnCrimssAppl.a	*CrIMSS Application library
libProCmnCrimssDictEntries.a, libProCmnCrimssDictEntries.so	*CrIMSS Dictionary Entries library
libProCmnCrisAlgorithmFactory.a, libProCmnCrisAlgorithmFactory.so	*PRO Common CrIS Algorithm Factory library
libProCmnDictEntries.so, libProCmnDictEntries.a	Common Dictionary Entries library
libProCmnDictionary.so, libProCmnDictionary.a	Common Dictionary library
libProCmnGeoloc.a	Common Geolocation library
libProCmnIPO.so, libProCmnIPO.a	Common Input Processing Output (IPO) library
libProCmnMapping.so, libProCmnMapping.a	Mapping library
libProCmnMath.so, libProCmnMath.a	Common Math library
libProCmnMode.so, libProCmnMode.a	Common Sensor Mode library
libProCmnOmpsNpAlgorithmFactory.a, libProCmnOmpsNpAlgorithmFactory.so	*PRO Common OMPS Nadir Profile Algorithm Factory library
libProCmnOmpsNpAppl.a, libProCmnOmpsNpAppl.so	*OMPS Nadir Profile Application library
libProCmnOmpsNpDictEntries.a, libProCmnOmpsNpDictEntries.so	*OMPS Nadir Profile Dictionary Entries library
libProCmnOmpsTcAlgorithmFactory.a, libProCmnOmpsTcAlgorithmFactory.so	*PRO Common OMPS Total Column Algorithm Factory library
libProCmnOmpsTcAppl.a, libProCmnOmpsTcAppl.so	*OMPS Total Column Application library
libProCmnOmpsTcDictEntries.a, libProCmnOmpsTcDictEntries.so	*OMPS Total Column Dictionary Entries library
libProCmnStrings.so, libProCmnStrings.a	Common Strings library
libProCmnUtil.so, libProCmnUtil.a	Common Utility library
libProCmnValidateDictionarySize.a, libProCmnValidateDictionarySize.so	Common Validate Dictionary Size library
libProCmnViirsAlgorithmFactory.a, libProCmnViirsAlgorithmFactory.so	*PRO Common VIIRS Algorithm Factory library

Library Name	Description
libProCmnViirsAppl.a, libProCmnViirsAppl.so	*VIIRS Application library
libProCmnViirsDictEntries.a, libProCmnViirsDictEntries.so	*VIIRS Dictionary Entries library
libProCmnViirsVerifiedRDRDictEntries.a, libProCmnViirsVerifiedRDRDictEntries.so	*VIIRS Verified RDR Dictionary Entries library
libProEdrCrimssAtmProf.so, libProEdrCrimssAtmProf.a	*CrIMSS EDR library
libProEdrCrimssAtmProfController.a, libProEdrCrimssAtmProfController.so	*CrIMSS EDR Controller library
libProEdrOmpsNP.a, libProEdrOmpsNP.so	*OMPS NP IP library
libProEdrOmpsNpController.a, libProEdrOmpsNpController.so	*OMPS NP IP Controller library
libProEdrOmpsNpGlueware.a, libProEdrOmpsNpGlueware.so	*OMPS NP IP Glueware library
libProEdrOmpsTc.a, libProEdrOmpsTc.so	*OMPS TC EDR science code library
libProEdrOmpsTcController.a, libProEdrOmpsTcController.so	*OMPS TC EDR Controller library
libProEdrOmpsTcEdr.a, libProEdrOmpsTcEdr.so	*OMPS TC EDR library
libProEdrOmpsTcIp.a, libProEdrOmpsTcIp.so	*OMPS TC IP library
libProEdrOmpsTcIpController.a, libProEdrOmpsTcIpController.so	*OMPS Total Column IP Controller library
libProEdrViirsActiveFires.so, libProEdrViirsActiveFires.a	*VIIRS Active Fires library
libProEdrViirsAerosol.so, libProEdrViirsAerosol.a	*VIIRS Aerosol EDR library
libProEdrViirsAerosolController.a, libProEdrViirsAerosolController.so	*VIIRS Aerosol Controller library
libProEdrViirsCM.a, libProEdrViirsCM.so	*VIIRS Cloud Mask IP library
libProEdrViirsClouds.so, libProEdrViirsClouds.a	*VIIRS Clouds library
libProEdrViirsCloudsFirstController.so, libProEdrViirsCloudsFirstController.a	*VIIRS Clouds First Controller library
libProEdrViirsCloudsSecondController.so, libProEdrViirsCloudsSecondController.a	*VIIRS Clouds Second Controller library
libProEdrViirsCop.so, libProEdrViirsCop.a	*VIIRS Cloud Optical Properties library
libProEdrViirsCopController.so, libProEdrViirsCopController.a	*VIIRS Cloud Optical Properties Controller library
libProEdrViirsIChannelImagery.so, libProEdrViirsIChannelImagery.a	*VIIRS I Channel Imagery EDR library
libProEdrViirsIST.so, libProEdrViirsIST.a	*VIIRS Ice Surface Temperature EDR library
libProEdrViirsIceAge.a, libProEdrViirsIceAge.so	*VIIRS Ice Age EDR library
libProEdrViirsIceAgeController.a, libProEdrViirsIceAgeController.so	*VIIRS Ice Age EDR Controller library
libProEdrViirsIceConc.a, libProEdrViirsIceConc.so	*VIIRS Ice Concentration EDR library
libProEdrViirsIceQual.so, libProEdrViirsIceQual.a	*VIIRS Ice Quality IP library
libProEdrViirsLST.a, libProEdrViirsLST.so	*VIIRS Land Surface Temperature EDR library
libProEdrViirsLsa.a, libProEdrViirsLsa.so	*VIIRS Land Surface Albedo EDR library
libProEdrViirsLsaController.a, libProEdrViirsLsaController.so	*VIIRS Land Surface Albedo EDR Controller library
libProEdrViirsLstController.a, libProEdrViirsLstController.so	*VIIRS Land Surface Temperature EDR Controller library
libProEdrViirsMChannelImagery.so, libProEdrViirsMChannelImagery.a	*VIIRS M Channel Imagery EDR library

Library Name	Description
libProEdrViirsMasksController.a, libProEdrViirsMasksController.so	*VIIRS Masks Controller library
libProEdrViirsNHF.a	*VIIRS NHF EDR library
libProEdrViirsNHFAlgo.so, libProEdrViirsNHFAlgo.a	*VIIRS NHF EDR Algorithm library
libProEdrViirsNHFGeo.so, libProEdrViirsNHFGeo.a	*VIIRS NHF Geolocation library
libProEdrViirsNHFutil_lib.a	*VIIRS NHF Utility library
libProEdrViirsNccImagery.so, libProEdrViirsNccImagery.a	*VIIRS NCC Imagery EDR library
libProEdrViirsNhfController.so, libProEdrViirsNhfController.a	*VIIRS NHF Controller library
libProEdrViirsOCC.so, libProEdrViirsOCC.a	*VIIRS OCC EDR library
libProEdrViirsOCCController.so, libProEdrViirsOCCController.a	*VIIRS OCC Controller library
libProEdrViirsSST.so, libProEdrViirsSST.a	*VIIRS SST EDR library
libProEdrViirsSnow.so, libProEdrViirsSnow.a	*VIIRS Snow EDR library
libProEdrViirsSstController.so, libProEdrViirsSstController.a	*VIIRS SST EDR Controller library
libProEdrViirsSurfReflect.so, libProEdrViirsSurfReflect.a	*VIIRS Surface Reflectance IP library
libProEdrViirsSurfReflectController.a, libProEdrViirsSurfReflectController.so	*VIIRS Surface Reflectance IP Controller library
libProEdrViirsSurfType.so, libProEdrViirsSurfType.a	*VIIRS Surface Type library
libProEdrViirsSurfTypeController.a, libProEdrViirsSurfTypeController.so	*VIIRS Surface Type Controller library
libProEdrViirsSurfaceAlbedo.a, libProEdrViirsSurfaceAlbedo.so	*VIIRS Surface Albedo library
libProEdrViirsVI.so, libProEdrViirsVI.a	*VIIRS Vegetation Index library
libProGip.a, libProGip.so	*PRO Gridding library
libProGipCrisController.a, libProGipCrisController.so	*CrIS GIP GridToGran Controller library
libProGipCrisGridToGran.a, libProGipCrisGridToGran.so	*CrIS GIP GridToGran library
libProGipCrisMapping.a, libProGipCrisMapping.so	*CrIS GIP GridToGran Mapping library
libProGipCsController.a, libProGipCsController.so	*Cross Sensor Controller library
libProGipCsGranToGrid.a, libProGipCsGranToGrid.so	*Cross Sensor GranToGrid library
libProGipCsTileInterfaces.a, libProGipCsTileInterfaces.so	*Cross Sensor Tile Interfaces library
libProGipMapping.a, libProGipMapping.so	*PRO Gridding Mapping library
libProGipViirs.a, libProGipViirs.so	*VIIRS Gridding library
libProGipViirsAW.a, libProGipViirsAW.so	*Area Weighting Gridding Granulation library
libProGipViirsDSRInitDispatcher.a, libProGipViirsDSRInitDispatcher.so	*VIIRS GIP Daily Surface Reflectance GridToGrid Dispatcher library
libProGipViirsGranToGridDSR.a, libProGipViirsGranToGridDSR.so	*VIIRS GIP Daily Surface Reflectance GranToGrid library
libProGipViirsGranToGridMonthlySRBTVI.a, libProGipViirsGranToGridMonthlySRBTVI.so	*VIIRS GIP Monthly SRBTVI GranToGrid library

Library Name	Description
libProGipViirsGranToGridSnowIceCover.a, libProGipViirsGranToGridSnowIceCover.so	*VIIRS GIP Snow Ice Cover GranToGrid library
libProGipViirsGridToGran.a, libProGipViirsGridToGran.so	*VIIRS Gridding to Granulation library
libProGipViirsGridToGrid.a, libProGipViirsGridToGrid.so	*VIIRS GIP GridToGrid library
libProGipViirsLSADispatcher.a, libProGipViirsLSADispatcher.so	*VIIRS GIP Land Surface Albedo GridToGrid Dispatcher library
libProGipViirsMonthlyInitDispatcher.a, libProGipViirsMonthlyInitDispatcher.so	*VIIRS GIP Monthly Initialization GridToGrid Dispatcher library
libProGipViirsMonthlyPostCompDispatcher.a, libProGipViirsMonthlyPostCompDispatcher.so	*VIIRS GIP Month Post Composite GridToGrid Dispatcher library
libProGipViirsNbarNdviController.a, libProGipViirsNbarNdviController.so	*VIIRS GIP NBAR NDVI GridToGrid Controller library
libProGipViirsNbarNdviDispatcher.a, libProGipViirsNbarNdviDispatcher.so	*VIIRS GIP NBAR NDVI GridToGrid Dispatcher library
libProGipViirsTileInterfaces.a, libProGipViirsTileInterfaces.so	Gridding VIIRS Tile Interface library
libProOmpsQuaternion.a, libProOmpsQuaternion.so	*OMPS Quaternion library
libProOmpsUtil.a, libProOmpsUtil.so	*OMPS Utility library
libProOmpsVerifiedRdr.so, libProOmpsVerifiedRdr.a	*OMPS Verified RDR library
libProScEAVerifiedRDRConverter.a, libProScEAVerifiedRDRConverter.so	Spacecraft Diary Verified RDR Converter library
libProSdrAtms.so, libProSdrAtms.a	*ATMS SDR library
libProSdrAtmsController.a, libProSdrAtmsController.so	*ATMS SDR Controller library
libProSdrAtmsRemap.a, libProSdrAtmsRemap.so	*ATMS SDR Remap library
libProSdrAtmsRemapController.a, libProSdrAtmsRemapController.so	*ATMS SDR Remap Controller library
libProSdrAtmsVerifiedRDR.so, libProSdrAtmsVerifiedRDR.a	*ATMS Verified RDR library
libProSdrCmnGeo.so, libProSdrCmnGeo.a	SDR Common Geolocation library
libProSdrCmnVerifiedRdr.so, libProSdrCmnVerifiedRdr.a	Common Verified RDR library
libProSdrCris.a, libProSdrCris.so	*CrIS SDR library
libProSdrCrisController.a, libProSdrCrisController.so	*CrIS SDR Controller library
libProSdrOmpsNp.a	*OMPS Nadir Profile library
libProSdrOmpsNpCal.a, libProSdrOmpsNpCal.so	*OMPS Nadir Profile Calibration library
libProSdrOmpsNpEarth.a, libProSdrOmpsNpEarth.so	*OMPS Nadir Profile Earth library
libProSdrOmpsNpEarthController.a, libProSdrOmpsNpEarthController.so	*OMPS Nadir Profile Earth Controller library
libProSdrOmpsTc.a	*OMPS Total Column library
libProSdrOmpsTcCal.a, libProSdrOmpsTcCal.so	*OMPS Total Column Calibration library
libProSdrOmpsTcEarth.so, libProSdrOmpsTcEarth.a	*OMPS Total Column Earth library
libProSdrOmpsTcEarthController.a, libProSdrOmpsTcEarthController.so	*OMPS Total Column Earth Controller library

Library Name	Description
libProSdrOmpsVerLut.a, libProSdrOmpsVerLut.so	*OMPS Version Look Up Table library
libProSdrTerrainCorrect.a	SDR Terrain Correction library
libProSdrViirs.a, libProSdrViirs.so	*VIIRS SDR library
libProSdrViirsBrightPixel.a, libProSdrViirsBrightPixel.so	*VIIRS SDR Bright Pixel library
libProSdrViirsCal.a, libProSdrViirsCal.so	*VIIRS SDR Calibration library
libProSdrViirsController.a, libProSdrViirsController.so	*VIIRS SDR Controller library
libProSdrViirsGeo.a, libProSdrViirsGeo.so	*VIIRS SDR Geolocation library
libProSdrViirsSolarDiffuser.a, libProSdrViirsSolarDiffuser.so	*VIIRS SDR Solar Diffuser library
libProSdrViirsVerifiedRdr.a, libProSdrViirsVerifiedRdr.so	*VIIRS SDR Verified RDR library
libProSrcAutogen.a	*PRO Source Autogeneration library
libnovasc.a	Naval Observatory Vector Astrometry Subroutines C Version library

* Indicates libraries only available after the IDPS Algorithms package has been extracted into the base ADL installation and compiled.

3.1.2 Executables

The following two tables list the ADL executables. These files are located in $ADL_HOME/bin or $ADL_HOME/tools/bin after a successful build. The executables listed in the table below are ADL compatible IDPS algorithms that can be run in the ADL environment. The executables listed in Table 3.1.2-2 are merely utilities that do not run any algorithm.

Table 3.1.2-1 Software Inventory – Algorithm Executables

Executable Name	Description
AdlEdrAlgorithmTemplate.exe	EDR Algorithm Template 1 Example
AdlEdrAlgorithmTemplate2.exe	EDR Algorithm Template 2 Example
AdlSdrAlgorithmTemplate.exe	SDR Algorithm Template Example
ProAncTemporalInterpolation.exe	*Temporal Interpolation Granulation
ProEdrCrimssAtmProfController.exe	*CrIMSS Atmosphere Profile EDR Controller
ProEdrOmpsNp.exe	*OMPS Nadir Profile IP Controller
ProEdrOmpsTcEdr.exe	*OMPS Total Column EDR Controller
ProEdrOmpsTcIp.exe	*OMPS Total Column IP Controller
ProEdrViirsAerosolController.exe	*VIIRS Aerosol EDR Controller
ProEdrViirsCloudsFirstController.exe	*VIIRS Cloud First Controller
ProEdrViirsCloudsSecondController.exe	*VIIRS Cloud Second Controller
ProEdrViirsCopController.exe	*VIIRS Cloud Optical Properties Controller
ProEdrViirsIChannelImagery.exe	*VIIRS I Channel Imagery
ProEdrViirsIceAgeController.exe	*VIIRS Ice Age Controller
ProEdrViirsIceConc.exe	*VIIRS Ice Concentratation
ProEdrViirsIceQual.exe	*VIIRS Ice Quality EDR
ProEdrViirsIst.exe	*VIIRS Ice Surface Temperature EDR
ProEdrViirsLsaController.exe	*VIIRS Land Surface Albedo EDR Controller
ProEdrViirsLstController.exe	*VIIRS Land Surface Temperature EDR Controller
ProEdrViirsMChannelImagery.exe	*VIIRS M Channel Imagery

Executable Name	Description
ProEdrViirsMasksController.exe	*VIIRS Masks Controller
ProEdrViirsNccImagery.exe	*VIIRS NCC Imagery
ProEdrViirsNhfController.exe	*VIIRS Net Heat Flux Controller
ProEdrViirsOCCController.exe	*VIIRS Ocean Color/Chlorophyll EDR Controller
ProEdrViirsSnow.exe	*VIIRS Snow Cover EDR
ProEdrViirsSstController.exe	*VIIRS Sea Surface Temperature EDR Controller
ProEdrViirsSurfReflectController.exe	*VIIRS Surface Reflectivity EDR Controller
ProEdrViirsSurfTypeController.exe	*VIIRS Surface Type EDR Controller
ProEdrViirsSurfaceAlbedo.exe	*VIIRS Surface Albedo EDR
ProEdrViirsVI.exe	*VIIRS Vegetation Index EDR
ProGipViirsDSRInitDispatcher.exe	*VIIRS GridToGrid Daily Surface Reflectance Initialization
ProGipViirsGranToGridDSR.exe	*VIIRS GranToGrid Daily Surface Reflectance
ProGipViirsGranToGridMonthlySRBTVI.exe	*VIIRS GranToGrid Monthly SRBTVI
ProGipViirsGranToGridSnowIceCover.exe	*VIIRS GranToGrid Snow Ice Cover
ProGipViirsLSADispatcher.exe	*VIIRS GridToGrid Daily Surface Reflectance Initialization
ProGipViirsMonthlyInitDispatcher.exe	*VIIRS GridToGrid Monthly Initialization
ProGipViirsMonthlyPostCompDispatcher.exe	*VIIRS GridToGrid Monthly Post Composite
ProGipViirsNbarNdviDispatcher.exe	*VIIRS GridToGrid NBAR NDVI
ProSdrAtmsController.exe	*ATMS SDR Controller
ProSdrAtmsRemapController.exe	*ATMS Remap Controller
ProSdrCrisController.exe	*CrIS SDR Controller
ProSdrOmpsNpCal.exe	*OMPS Nadir Profile Calibration
ProSdrOmpsNpEarth.exe	*OMPS Nadir Profile Earth Controller
ProSdrOmpsTcCal.exe	*OMPS Total Column Calibration
ProSdrOmpsTcEarth.exe	*OMPS Total Column Earth
ProSdrViirsBrightPixel.exe	*VIIRS Bright Pixel SDR
ProSdrViirsController.exe	*VIIRS SDR Controller
ProSdrViirsSolarDiffuser.exe	*VIIRS Solar Diffuser SDR

* Indicates executables only available after the IDPS Algorithms package has been extracted into the base ADL installation and compiled.

Table 3.1.2-2 Software Inventory – Tool Executables

Executable Name	Description
ADL_Packer.exe	Utility to pack a binary file into an HDF5 file
ADL_Unpacker.exe	Utility to unpack a binary file from an HDF5 file
AdlCmnGenerateDerivedAlgorithm.exe	ADL Derived Algorithm Class Generator Utility
AdlFileEndianConverter.exe	File Endian Converter Stand Alone Utility
CrisGenerateTelemetryDataFile.exe	*CrIS Telemetry Data File Generator Utility
InfUtil_PerfFormat.exe	INF Performance Format Utility
IngMsd.exe	ING Mission Support Data Utility
ProCmnGenerateDerivedAlgorithm.exe	*PRO Derived Algorithm Class Generator Utility
UridCreator.exe	URID Creation Utility
float	Converts IEEE integer formatted numbers to floating point numbers.
granID.exe	NPP Granule ID Utility
string2cds.exe	CCSDS Time Conversion Utility
string2iet.exe	IET Time Conversion Utility

* Indicates executables only available after the IDPS Algorithms package has been extracted into the base ADL installation and compiled.

3.1.3 Guide List Files

Each ADL algorithm process has a guide list file which specifies all of the configuration files that are needed by the process to execute successfully. The ADL guide list files for the ADL compatible IDPS algorithms are listed in the following table and there is a one to one relationship with the ADL executables. The entire ADL guide lists are located in $ADL_HOME/cfg.

Table 3.1.3-1 Guide List Files

File Name	Description
AdlEdrAlgorithmTemplate_GuideList.cfg	AdlEdrAlgorithmTemplate.exe guide list
AdlEdrAlgorithmTemplate2_GuideList.cfg	AdlEdrAlgorithmTemplate2.exe guide list
AdlSdrAlgorithmTemplate_GuideList.cfg	AdlSdrAlgorithmTemplate.exe guide list
DDS_ASC_GuideList.cfg	ADL_Unpacker.exe guide list
DDS_DDSUTIL_GuideList.cfg	ADL_Packer.exe guide list
IngMsd_GuideList.cfg	ADL MSD guide list
ProAncTemporalInterpolation_GuideList.cfg	*ProAncTemporalInterpolation.exe guide list
ProEdrCrimssAtmProfController_GuideList.cfg	*ProEdrCrimssAtmProfController.exe guide list
ProEdrOmpsNp_GuideList.cfg	*ProEdrOmpsNp.exe guide list
ProEdrOmpsTcEdr_GuideList.cfg	*ProEdrOmpsTcEdr.exe guide list
ProEdrOmpsTcIp_GuideList.cfg	*ProEdrOmpsTcIp.exe guide list
ProEdrViirsAerosolController_GuideList.cfg	*ProEdrViirsAerosolController.exe guide list
ProEdrViirsCloudsFirstController_GuideList.cfg	*ProEdrViirsCloudsFirstController.exe guide list
ProEdrViirsCloudsSecondController_GuideList.cfg	*ProEdrViirsCloudsSecondController.exe guide list
ProEdrViirsCopController_GuideList.cfg	*ProEdrViirsCopController.exe guide list
ProEdrViirsGtmIChannelImagery_GuideList.cfg	*ProEdrViirsIChannelImagery.exe guide list
ProEdrViirsGtmMChannelImagery_GuideList.cfg	*ProEdrViirsMChannelImagery.exe guide list
ProEdrViirsGtmNccImagery_GuideList.cfg	*ProEdrViirsNccImagery.exe guide list
ProEdrViirsIceAgeController_GuideList.cfg	*ProEdrViirsIceAgeController.exe guide list
ProEdrViirsIceConc_GuideList.cfg	*ProEdrViirsIceConc.exe guide list
ProEdrViirsIceQual_GuideList.cfg	*ProEdrViirsIceQual.exe guide list
ProEdrViirsIceSurfTemp_GuideList.cfg	*ProEdrViirsIst.exe guide list
ProEdrViirsLsaController_GuideList.cfg	*ProEdrViirsLsaController.exe guide list
ProEdrViirsLstController_GuideList.cfg	*ProEdrViirsLstController.exe guide list
ProEdrViirsMasksController_GuideList.cfg	*ProEdrViirsMasksController.exe guide list
ProEdrViirsNhfController_GuideList.cfg	*ProEdrViirsNhfController.exe guide list
ProEdrViirsOCCController_GuideList.cfg	*ProEdrViirsOCCController.exe guide list
ProEdrViirsSSTController_GuideList.cfg	*ProEdrViirsSstController.exe guide list
ProEdrViirsSnow_GuideList.cfg	*ProEdrViirsSnow.exe guide list
ProEdrViirsSurfReflectController_GuideList.cfg	*ProEdrViirsSurfReflectController.exe guide list
ProEdrViirsSurfTypeController_GuideList.cfg	*ProEdrViirsSurfTypeController.exe guide list
ProEdrViirsSurfaceAlbedo_GuideList.cfg	*ProEdrViirsSurfaceAlbedo.exe guide list
ProEdrViirsVI_GuideList.cfg	*ProEdrViirsVI.exe guide list
ProGipViirsDSRInitDispatcher_GuideList.cfg	*ProGipViirsDSRInitDispatcher.exe guide list
ProGipViirsGranToGridDSR_GuideList.cfg	*ProGipViirsGranToGridDSR.exe guide list
ProGipViirsGranToGridMonthlySRBTVI_GuideList.cfg	*ProGipViirsGranToGridMonthlySRBTVI.exe guide list

File Name	Description
ProGipViirsGranToGridSnowIceCover_GuideList.cfg	*ProGipViirsGranToGridSnowIceCover.exe guide list
ProGipViirsLSADispatcher_GuideList.cfg	*ProGipViirsLSADispatcher.exe guide list
ProGipViirsMonthlyInitDispatcher_GuideList.cfg	*ProGipViirsMonthlyInitDispatcher.exe guide list
ProGipViirsMonthlyPostCompDispatcher_GuideList.cfg	*ProGipViirsMonthlyPostCompDispatcher.exe guide list
ProGipViirsNbarNdviDispatcher_GuideList.cfg	*ProGipViirsNbarNdviDispatcher.exe guide list
ProSdrAtmsRemapController_GuideList.cfg	*ProSdrAtmsRemapController.exe guide list
ProSdrAtms_GuideList.cfg	*ProSdrAtmsController.exe guide list
ProSdrCris_GuideList.cfg	*ProSdrCrisController.exe guide list
ProSdrOmpsNpCal_GuideList.cfg	*ProSdrOmpsNpCal.exe guide list
ProSdrOmpsNpEarth_GuideList.cfg	*ProSdrOmpsNpEarth.exe guide list
ProSdrOmpsTcCal_GuideList.cfg	*ProSdrOmpsTcCal.exe guide list
ProSdrOmpsTcEarth_GuideList.cfg	*ProSdrOmpsTcEarth.exe guide list
ProSdrViirsBrightPixel_GuideList.cfg	*ProSdrViirsBrightPixel.exe guide list
ProSdrViirsSolarDiffuser_GuideList.cfg	*ProSdrViirsSolarDiffuser.exe guide list
ProSdrViirs_GuideList.cfg	*ProSdrViirsController.exe guide list

* Indicates guide lists only available after the IDPS Algorithms package has been extracted into the base ADL installation.

3.2 Software Environment

ADL is supported on x86-64 systems running the Red Hat Enterprise Linux operating system and POWER systems running the AIX operating system. See the table below for specific OS version support. The ADL source code needs to be compiled with either the IBM XL C/C++/Fortran or GCC compilers listed in the below table.

Table 3.2-1 Compatible OS and Compilers

OS	C compiler	C++ compiler	Fortran 90 compiler
AIX 5.3.0.0 or later	IBM XL C/C++ Enterprise Edition for AIX, V10.1	IBM XL C/C++ Enterprise Edition for AIX, V10.1	IBM XL Fortran Enterprise Edition for AIX, V12.1
LINUX (Red Hat Enterprise 5)	GCC 4.4.0	GCC 4.4.0	GCC 4.4.0

3.2.1 COTS Products

ADL COTS software requirements are listed in the table below. This software needs to be installed prior to compilation of the ADL software.

Table 3.2.1-1 ADL COTS Software

Product	Company	Platform	Version	Why needed
Xerces - C++ XML Parser	The Apache Software Foundation	AIX, Linux	2.6.0	Re-use the INF config guide reader utility software as operational software. Autogenerate CrIS SDR code
Java SE 6	Oracle	AIX, Linux	Java 1.6	XML GUI Editor tool

Product	Company	Platform	Version	Why needed
Perl	Open Source	AIX, Linux	perl-5.8.1	Used for scripting auto generation software
Perl XML::Simple, Getopt::Std, File::Basename, Data::Dumper, FileHandle, Exporter modules all available for download.	Open Source	AIX, Linux	perl-5.8.1	Used for scripting auto generation software
Imake	Open Source	AIX, Linux	1.0.2-3	Used to build ADL software
log4j JAR	Apache	AIX, Linux	1.2.8	Used for logging in the XML Editor tool
Ant -Java-based build tool	Apache	AIX, Linux	1.6.5	Used for building the Java code
Boost	Open Source	AIX, Linux	1.32.0	Used in VIIRS SDR, EDR, and GIP, CrIS CrIMSS, ATMS, ATMSRemap, and OMPS SDR, CrIMSS EDR, CMN, Geolocation, xml, ANC OMPS,
wgrib2	http://www.cpc.ncep.noaa.gov/products/wesley/wgrib2/	AIX, Linux	0.1.4e	Used by ING MSD
jasper	http://www.ece.uvic.ca/~frodo/jasper/	AIX, Linux	1.700.5	Used by ING MSD
g2clib	http://www.nco.ncep.noaa.gov/pmb/docs/grib2/download/	AIX, Linux	1.0.4	Used by ING MSD
HDF5	The HDF Group	AIX, Linux	1.8.6	Used for packaging and unpacking CDFCB compliant HDF files, CMN
Log4cplus	Open Source	AIX, Linux	1.0.2 (Version 1.0.4 will not work)	Used by the INF Debug Logger utility.

IDPS Algorithm COTS software requirements are listed in the table below. This software needs to be installed prior to compilation of the IDPS Algorithm software.

Table 3.2.1-2 IDPS Algorithm COTS Software

Product	Company	Platform	Version	Why needed
DCDFLIB	Netlib	AIX, Linux	1.1 (C language version. Use of the Fortran version requires minor code	Used by VIIRS Solar Diffuser

Product	Company	Platform	Version	Why needed
			changes.)	
BLAS	Netlib	AIX, Linux	3.0	Used by VIIRS, ATMS, CrIS, ATMSRemap SDR, VIIRS and CrIMSS EDR for math operations
FFTW	Open Source	AIX, Linux	3.0.1	Used by CrIS, ATMS, ATMSRemap, VIIRS SDR, and CrIMSS EDR for math operations
LAPACK	Netlib	AIX, Linux	3.0	Used by VIIRS, ATMS, ATMSRemap, and CrIS SDR, VIIRS GIP, and VIIRS and CrIMSS EDR for math operations
PPPACK	Netlib	AIX, Linux	8/18/2004	Used by OMPS, CrIS, ATMS, ATMSRemap, and VIIRS SDR, OMPS and VIIRS EDR, VIIRS GIP, Geolocation, and CMN for math operations
SLATEC	Netlib	AIX, Linux	4.1	Used by OMPS SDR for math operations

3.2.2 Environment Variables

The ADL software needs the following environment variables set prior to compiling the ADL software in order to know where COTS software is installed. Example envSetup.ksh/envSetup.csh scripts are included with the ADL software and can be found under $ADL_HOME/build. These setup scripts will need to be modified to change the environment variables below to the appropriate locations of the installed COTS software. The following table lists environment variables that will need to be modified for ADL to compile. Some additional environment variables are set in envSetup.ksh/envSetup.csh, but should be set automatically and are not listed in the table.

Table 3.2.2-1 Environment Variables

Variable Name	Value
ADL_HOME	Set to the root directory of the ADL software installation location
ADL_MSD_LZ_PATH	Directory of the ADL MSD Landing Zone
ANT_HOME	ANT home installation directory
COTS_BLAS_LIB_LOC	Directory containing the BLAS library
COTS_BOOST_FILESYSTEM_LIB	Library name of the BOOST file system library
COTS_BOOST_INC	Boost installation directory
COTS_BOOST_LIB	Boost library directory
COTS_BOOST_REGEX_LIB	Library name of the BOOST regular expression library
COTS_BOOST_SERIAL_LIB	Library name of the BOOST serialization library
COTS_DCDFLIB_LIB_LOC	Directory containing the DCDFLIB library
COTS_FFTW_HOME	FFTW installation directory
COTS_HDF5_HOME	HDF5 installation directory

COTS_LAPACK_LIB_LOC	Directory containing the LAPACK library
COTS_LOG4CPLUS_HOME	Log4cplus installation directory
COTS_PPPACK_LIB_LOC	Directory containing the PPPACK library
COTS_SLATEC_LIB_LOC	Directory containing the SLATEC library
COTS_WGRIB2_HOME	WGRIB installation direcotory
COTS_XER_CPP_HOME	Xerces - C++ XML Parser home installation directory
DCONFIG	Directory containing the algorithm config guide and guide list files
ENDIAN	Endianness of the host, either “BIG” or “LITTLE”
JAVA_HOME	Java 6 64 bit installation directory
LOG4J_JAR_FILE	Log4j jar file installation directory
NPP_GRANULE_ID_BASETIME	This is automatically set correctly for data which ships with ADL and for proxy data. However, it will need to be updated if ADL is used with real data from the NPP spacecraft.

The ADL build software also needs locations to some Unix commands and the compiler being used (GCC/XL). The proper locations should be updated in the $ADL_HOME/imakeconf/IDPS_Conf_Params file. There is a section for GCC and XL compilers in this file, the commands need to be updated for the proper tool install location.

3.2.3 Data Sources

Several data packages are delivered along with the ADL software packages and the IDPS algorithm software package. These data packages contain NPP proxy test data for demonstrating the ADL compatible IDPS algorithms and truth data produced on an AIX system by the IDPS operational software. This data can be found in $ADL_HOME/data/input and $ADL_HOME/data/output. This data is purely for running the ADL compatible IDPS algorithms only and is not required when running a new algorithm in the ADL framework.

Inside of the input directory are two more directories: noMetadata and withMetadata. Each of those directories contains numerous directories, one for each ADL algorithm. The data inside of the noMetadata directory contains data that can be used to run an algorithm in “no metadata” mode. The data inside of the withMetadata directory contains data to run an algorithm with metadata.

The output directory also contains noMetadata and withMetadata directories. Inside each of those are two directories for each algorithm: <algorithm>Outputs and <algorithm>TruthOutputs. When an algorithm is executed in ADL, the default configured location to write outputs is the <algorithm>Outputs directory. Outputs from running the operational algorithms are provided as truth data in <algorithm>TruthOutputs.

The directories inside of $ADL_HOME/data make extensive use of symbolic links to reduce the size of the provided test data.

3.2.4 Configuration Files

ADL uses the same algorithm configuration files as PRO in IDPS to specify the inputs and outputs to an algorithm. The only distinction between ADL algorithm configuration files and

PRO configuration files is that the collection short name entries under the inputs and outputs are the name of the data file if running in “no metadata” mode. However, if the data file name is the same as the actual collection short name, then the files are exactly the same. The input section of the configuration file identifies the primary, secondary, alternate and conversion inputs to an algorithm. The output products that will be produced by the algorithm are identified in the output section of the configuration files. Refer to PRO SW Configuration Guide User’s Manual for more information on the functionality of the configuration guides and the format.

The ADL configuration files are listed in the table below and stored in $ADL_HOME/cfg.

Table 3.2.4-1 Configuration Files

File Name	Description
ADL_CFG.xml	ADL parameters configuration guide
ADL_CFG_MSD.xml	ADL specific ING MSD configuration guide
ANC_CFG_Shared.xml	Shared configuration guide between ING MSD and PRO Temporal Interpolation
AdlEdrAlgorithmTemplate_CFG.xml	EDR Algorithm Template 1 configuration guide
AdlEdrAlgorithmTemplate2_CFG.xml	EDR Algorithm Template 2 configuration guide
AdlSdrAlgorithmTemplate_CFG.xml	SDR Algorithm Template configuration guide
DDSADL_CFG.xml	DDS HDF Unpacker configuration guide
DDSRH_CFG.xml	DDS HDF Packer configuration guide
DDSRH_LW_CFG.xml	DDS HDF Packer configuration guide
DICTIONARY_FILL_UT_CFG.xml	*Product Dictionary Fill Unit Test configuration guide
DICTIONARY_SCALEOFFSET_UT_CFG.xml	*Product Dictionary Scaling Unit Test configuration guide
DMS_CFG.xml	Data Management System configuration guide
DPE_CFG.xml	Data Processing Environment configuration guide
INF_CFG.xml	INF configuration guide
INF_CLOG_CFG.xml	INF Calculate List of Granules configuration guide
ING_CFG_Common.xml	ING Common configuration guide
PRO_ANC_CFG.xml	ANC Temporal Interpolation configuration guide
PRO_CMN_ADJ_CFG.xml	Common Adjacency configuration guide
PRO_CMN_GEO_CFG.xml	Common Geolocation configuration guide
PRO_MODE_MANEUVER_CFG.xml	*Mode and Maneuver configuration guide
ProAncCrisController_CFG.xml	*CrIS Ancillary Granulation Controller configuration guide
ProAncCrisGranulatePresLevelTemp_CFG.xml	*CrIS Temperature at Pressure Levels Granulation configuration guide
ProAncCrisGranulateSpecSurfHumidity_CFG.xml	*CrIS Specific Humidity Granulation configuration guide

File Name	Description
ProAncCrisGranulateSurfGeopotentialHeight_CFG.xml	*CrIS Surface Geopotential Height Granulation configuration guide
ProAncCrisGranulateSurfPres_CFG.xml	*CrIS Surface Pressure Granulation configuration guide
ProAncCrisGranulateSurfTemp_CFG.xml	*CrIS Surface Temperature Granulation configuration guide
ProAncCrisGranulateTerrainGeopotentialHeight_CFG.xml	*CrIS Terrain Geopotential Height Granulation configuration guide
ProAncCrisGranulateWaterVaporMixRatio_CFG.xml	*CrIS Water Vapor Mixing Ratio Granulation configuration guide
ProAncCrisPerformSurfPresCorrection_CFG.xml	*CrIS Surface Pressure Correction configuration guide
ProAncOmpsNpController_CFG.xml	*OMPS Nadir Profile Ancillary Granulation Controller configuration guide
ProAncOmpsTcController_CFG.xml	*OMPS Total Column Ancillary Granulation Controller configuration guide
ProAncOmpsTcGranulateCloudTopPres_CFG.xml	*OMPS Total Column Cloud Top Pressure Granulation configuration guide
ProAncOmpsTcGranulatePresLevelTemp_CFG.xml	*OMPS Total Column Temperature at Pressure Levels Granulation configuration guide
ProAncOmpsTcGranulateSurfPres_CFG.xml	*OMPS Total Column Surface Pressure Granulation configuration guide
ProAncOmpsTcGranulateUVSurfReflect_CFG.xml	*OMPS Total Column Ultra Violet Surface Reflectance Granulation configuration guide
ProAncTemporalInterpolation_CFG.xml	*Ancillary Data Temporal Interpolation configuration guide
ProAncViirsController_CFG.xml	*VIIRS Ancillary Controller configuration guide
ProAncViirsGranulateAotClimatology_CFG.xml	*VIIRS AOT Climatology Granulation configuration guide
ProAncViirsGranulateBathymetry_CFG.xml	*VIIRS Bathymetry Granulation configuration guide
ProAncViirsGranulateGeopotentialHeight_CFG.xml	*VIIRS Geopotential Height Granulation configuration guide
ProAncViirsGranulateNhfOzone_CFG.xml	*VIIRS Net Heat Flux Ozone Granulation configuration guide
ProAncViirsGranulateNhfPresLevelTemp_CFG.xml	*VIIRS Net Heat Flux Temperature at Pressure Levels Granulation configuration guide
ProAncViirsGranulateNhfSpecSurfHumidity_CFG.xml	*VIIRS Net Heat Flux Specific Humidity Granulation configuration guide
ProAncViirsGranulateNhfSurfPres_CFG.xml	*VIIRS Net Heat Flux Surface Pressure Granulation configuration guide
ProAncViirsGranulateNhfSurfTemp_CFG.xml	*VIIRS Net Heat Flux Surface Temperature Granulation configuration guide
ProAncViirsGranulateNhfWaterVaporMixRatio_CFG.xml	*VIIRS Net Heat Flux Water Vapor Mixing Ratio Granulation configuration guide
ProAncViirsGranulateNitrateDepletion_CFG.xml	*VIIRS Nitrate Depletion Temperature Granulation configuration guide

File Name	Description
ProAncViirsGranulateOpticalDepth_CFG.xml	*VIIRS Optical Depth Granulation configuration guide
ProAncViirsGranulateOzone_CFG.xml	*VIIRS Ozone Granulation configuration guide
ProAncViirsGranulatePrecipWater_CFG.xml	*VIIRS Precipitable Water Granulation configuration guide
ProAncViirsGranulatePresLevelTemp_CFG.xml	*VIIRS Temperature at Pressure Levels Granulation configuration guide
ProAncViirsGranulateSkinTemp_CFG.xml	*VIIRS Skin Temperature Granulation configuration guide
ProAncViirsGranulateSpecSurfHumidity_CFG.xml	*VIIRS Specific Humidity at the Surface Granulation configuration guide
ProAncViirsGranulateSurfGeopotentialHeight_CFG.xml	*VIIRS Surface Geopotential Height Granulation configuration guide
ProAncViirsGranulateSurfPres_CFG.xml	*VIIRS Surface Pressure Granulation configuration guide
ProAncViirsGranulateSurfTemp_CFG.xml	*VIIRS Surface Temperature Granulation configuration guide
ProAncViirsGranulateTerrainGeopotentialHeight_CFG.xml	*VIIRS Terrain Geopotential Height configuration guide
ProAncViirsGranulateTropoGeopotentialHeight_CFG.xml	*VIIRS Geopotential Height of the Troposphere Granulation configuration guide
ProAncViirsGranulateWaterVaporMixRatio_CFG.xml	*VIIRS Water Vapor Mixing Ratio Granulation configuration guide
ProAncViirsGranulateWindDirection_CFG.xml	*VIIRS Wind Direction Granulation configuration guide
ProAncViirsGranulateWindSpeed_CFG.xml	*VIIRS Wind Speed Granulation configuration guide
ProAncViirsPerformSurfPresCorrection_CFG.xml	*VIIRS Surface Pressure Correction configuration guide
ProCSMatchupCrimssGranulesToOmpsTc_CFG.xml	*CrIMSS to OMPS TC Granule Matchup configuration guide
ProCSMatchupViirsGranulesToOmpsTc_CFG.xml	*VIIRS to OMPS TC Matchup configuration guide
ProCmnCompareProducts_CFG.xml	*Product Dictionary Validation configuration guide
ProCmnProductDictionary_CFG.xml	Product Dictionary configuration guide
ProEdrCrimssAtmProfController_CFG.xml	*CrIMSS EDR Controller configuration guide
ProEdrCrimssAtmProf_CFG.xml	*CrIMSS EDR configuration guide
ProEdrOmpsNP_CFG.xml	*OMPS Nadir Profile IP configuration guide
ProEdrOmpsNpController_CFG.xml	*OMPS Nadir Profile IP Controller configuration guide
ProEdrOmpsNpGlueware_CFG.xml	*OMPS Nadir Profile Glueware configuration guide
ProEdrOmpsTcController_CFG.xml	*OMPS Total Column EDR Controller configuration guide
ProEdrOmpsTcEdr_CFG.xml	*OMPS Total Column EDR configuration guide
ProEdrOmpsTcIpController_CFG.xml	*OMPS Total Column IP Controller configuration guide

File Name	Description
ProEdrOmpsTcIp_CFG.xml	*OMPS Total Column IP configuration guide
ProEdrViirsActiveFires_CFG.xml	*VIIRS Active Fires configuration guide
ProEdrViirsAerosolController_CFG.xml	*VIIRS Aerosol EDR Controller configuration guide
ProEdrViirsAerosol_CFG.xml	*VIIRS Aerosol EDR configuration guide
ProEdrViirsCM_CFG.xml	*VIIRS Cloud Mask EDR configuration guide
ProEdrViirsCbh_CFG.xml	*VIIRS Cloud Base Height EDR configuration guide
ProEdrViirsCcl_CFG.xml	*VIIRS Cloud Cover Layers EDR configuration guide
ProEdrViirsCloudsFirstController_CFG.xml	*VIIRS Clouds EDR Controller #1 configuration guide
ProEdrViirsCloudsSecondController_CFG.xml	*VIIRS Clouds EDR Controller #2 configuration guide
ProEdrViirsCopController_CFG.xml	*VIIRS Cloud Optical Properties EDR Controller configuration guide
ProEdrViirsCop_CFG.xml	*VIIRS Cloud Optical Properties EDR configuration guide
ProEdrViirsCtp_CFG.xml	*VIIRS Cloud Top Parameters EDR configuration guide
ProEdrViirsGce_CFG.xml	*VIIRS Generate Cloud EDRs EDR configuration guide
ProEdrViirsIChannelImagery_CFG.xml	*VIIRS Ground Track Mercator I-band EDR configuration guide
ProEdrViirsIceAgeController_CFG.xml	*VIIRS Ice Age EDR Controller configuration guide
ProEdrViirsIceAge_CFG.xml	*VIIRS Ice Age EDR configuration guide
ProEdrViirsIceConc_CFG.xml	*VIIRS Ice Concentration EDR configuration guide
ProEdrViirsIceQual_CFG.xml	*VIIRS Ice Quality EDR configuration guide
ProEdrViirsIceSurfTemp_CFG.xml	*VIIRS Ice Surface Temperature EDR configuration guide
ProEdrViirsLandSurfAlbedo_CFG.xml	*VIIRS Land Surface Albedo EDR configuration guide
ProEdrViirsLsaController_CFG.xml	*VIIRS Land Surface Albedo EDR Controller configuration guide
ProEdrViirsLstController_CFG.xml	*VIIRS Land Surface Temperature EDR Controller configuration guide
ProEdrViirsLst_CFG.xml	*VIIRS Land Surface Temperature EDR configuration guide
ProEdrViirsMChannelImagery_CFG.xml	*VIIRS Ground Track Mercator M-band EDR configuration guide
ProEdrViirsMasksController_CFG.xml	*VIIRS Masks Controller configuration guide
ProEdrViirsNHFGeo_CFG.xml	*VIIRS Net Heat Flux EDR Geolocation configuration guide
ProEdrViirsNHF_CFG.xml	*VIIRS Net Heat Flux EDR configuration guide
ProEdrViirsNccImagery_CFG.xml	*VIIRS Near Constant Contrast Imagery EDR configuration guide

File Name	Description
ProEdrViirsNhfController_CFG.xml	*VIIRS Net Heat Flux EDR Controller configuration guide
ProEdrViirsOCC_CFG.xml	*VIIRS Ocean Color Chlorophyll EDR configuration guide
ProEdrViirsOccController_CFG.xml	*VIIRS Ocean Color Chlorophyll EDR Controller configuration guide
ProEdrViirsPpc_CFG.xml	*VIIRS Perform Parallax Correction EDR configuration guide
ProEdrViirsSnow_CFG.xml	*VIIRS Snow Cover Depth EDR configuration guide
ProEdrViirsSstController_CFG.xml	*VIIRS Sea Surface Temperature EDR Controller configuration guide
ProEdrViirsSst_CFG.xml	*VIIRS Sea Surface Temperature EDR configuration guide
ProEdrViirsSurfReflectController_CFG.xml	*VIIRS Surface Reflectance EDR Controller configuration guide
ProEdrViirsSurfReflect_CFG.xml	*VIIRS Surface Reflectance EDR configuration guide
ProEdrViirsSurfTypeController_CFG.xml	*VIIRS Surface Type EDR Controller configuration guide
ProEdrViirsSurfType_CFG.xml	*VIIRS Surface Type EDR configuration guide
ProEdrViirsSurfaceAlbedo_CFG.xml	*VIIRS Surface Albedo EDR configuration guide
ProEdrViirsVI_CFG.xml	*VIIRS Vegetative Index EDR configuration guide
ProGipCSGranToGridCrimssController_CFG.xml	*CrIMSS GranToGrid Cross Sensor Controller configuration guide
ProGipCSGranToGridCrimssTempProfile_CFG.xml	*CrIMSS Temperature Profile GranToGrid Cross Sensor configuration guide
ProGipCSGranToGridViirsCloudTopPres_CFG.xml	*VIIRS Cloud Top Pressure GranToGrid Cross Sensor configuration guide
ProGipCSGranToGridViirsSnowIce_CFG.xml	*VIIRS Snow Ice Cover GranToGrid Cross Sensor configuration guide
ProGipCSOmpsTcController_CFG.xml	*OMPS Total Column Cross Sensor Controller configuration guide
ProGipCSOmpsTcGranToGridCrimssDispatcher_CFG.xml	*CrIMSS GranToGrid Cross Sensor Dispatcher configuration guide
ProGipCSOmpsTcGranToGridViirsCtpDispatcher_CFG.xml	*VIIRS Cloud Top Parameters GranToGrid Cross Sensor Dispatcher configuration guide
ProGipCSOmpsTcGranToGridViirsSnowIceDispatcher_CFG.xml	*VIIRS Snow Ice Cover GranToGrid Cross Sensor configuration guide
ProGipCSOmpsTcGridToGranCloudTopParms_CFG.xml	*OMPS Total Column Cloud Top Parameters GridToGran Cross Sensor configuration guide
ProGipCSOmpsTcGridToGranSnowIce_CFG.xml	*OMPS Total Column Snow Ice Cover GridToGran Cross Sensor configuration guide
ProGipCSOmpsTcGridToGranTempProfile_CFG.xml	*OMPS Total Column Cross Sensor GridToGran For CrIMSS Temperature Profile configuration guide

File Name	Description
ProGipCris5kGridToGranMapping_CFG.xml	*CrIS GridToGran 5km Mapping configuration guide
ProGipCrisController_CFG.xml	*CrIS Gridded IP Controller configuration guide
ProGipCrisGranTo5kGridMapping_CFG.xml	*CrIS GranToGrid 5km Mapping configuration guide
ProGipCrisGridToGranLandFraction_CFG.xml	*CrIS GridToGran Land Fraction configuration guide
ProGipCrisGridToGranMapping_CFG.xml	*CrIS GridToGran Mapping configuration guide
ProGipOmpsNpGridToGranMapping_CFG.xml	*OMPS Nadir Profile GridToGran Mapping configuration guide
ProGipOmpsNpGridToGranSunGlint_CFG.xml	*OMPS Nadir Profile GridToGran Sun Glint configuration guide
ProGipOmpsTcGridToGranMapping_CFG.xml	*OMPS Total Column GridToGran Mapping configuration guide
ProGipOmpsTcGridToGranSnowIceFraction_CFG.xml	*OMPS Total Column GridToGran Snow Ice Fraction configuration guide
ProGipOmpsTcGridToGranSunGlint_CFG.xml	*OMPS Total Column GridToGran Sun Glint configuration guide
ProGipViirsDSRInitDispatcher_CFG.xml	*VIIRS Daily Surface Reflectance Initialization configuration guide
ProGipViirsGranToGridDSR_CFG.xml	*VIIRS GranToGrid Daily Surface Reflectance configuration guide
ProGipViirsGranToGridMonthlySRBTVI_CFG.xml	*VIIRS GranToGrid Monthly SRBTVI configuration guide
ProGipViirsGranToGridSnowIceCover_CFG.xml	*VIIRS GranToGrid Snow Ice Cover configuration guide
ProGipViirsGridToGranController_CFG.xml	*VIIRS GridToGran Controller configuration guide
ProGipViirsGridToGranLandSurfAlbedo_CFG.xml	*VIIRS GridToGran Land Surface Albedo configuration guide
ProGipViirsGridToGranMaxMinNdvi_CFG.xml	*VIIRS GridToGran Max Min NDVI configuration guide
ProGipViirsGridToGranNbarNdvi_CFG.xml	*VIIRS GridToGran NBAR NDVI Granulation configuration guide
ProGipViirsGridToGranQstLwm_CFG.xml	*VIIRS GridToGran Quarterly Surface Type Land Water Mask Granulation configuration guide
ProGipViirsGridToGranQst_CFG.xml	*VIIRS GridToGran Quarterly Surface Type Granulation configuration guide
ProGipViirsGridToGranSnowIceCover_CFG.xml	*VIIRS GridToGran Granulation Snow Ice Cover configuration guide
ProGipViirsGridToGrid17DayNbarNdvi_CFG.xml	*VIIRS GridToGrid 17 Day NBAR NDVI configuration guide
ProGipViirsGridToGridDSRInit_CFG.xml	*VIIRS GridToGrid Daily Surface Reflectance Initialization configuration guide
ProGipViirsGridToGridLSA_CFG.xml	*VIIRS GridToGrid Land Surface Albedo configuration guide
ProGipViirsGridToGridMonthlyInit_CFG.xml	*VIIRS GridToGrid Monthly Tile Initialization configuration guide

File Name	Description
ProGipViirsGridToGridMonthlyNbarNdvi_CFG.xml	*VIIRS GridToGrid Monthy NBAR NDVI configuration guide
ProGipViirsGridToGridMonthlyPostComp_CFG.xml	*VIIRS GridToGrid Montly Post Composite configuration guide
ProGipViirsGridToGridRollingNbarNdvi_CFG.xml	*VIIRS GridToGrid Rolling NBAR NDVI configuration guide
ProGipViirsLSADispatcher_CFG.xml	*VIIRS Land Surface Albedo Dispatcher configuration guide
ProGipViirsMonthlyInitDispatcher_CFG.xml	*VIIRS Monthly Tile Initialization Dispatcher configuration guide
ProGipViirsMonthlyPostCompDispatcher_CFG.xml	*VIIRS Monthly Post Composite Dispatcher configuration guide
ProGipViirsNNImg_CFG.xml	*VIIRS Nearest Neighbor Mapping IP configuration guide
ProGipViirsNbarNdviController_CFG.xml	*VIIRS NBAR NDVI Controller configuration guide
ProGipViirsNbarNdviDispatcher_CFG.xml	*VIIRS NBAR NDVI Dispatcher configuration guide
ProGipViirsWCalcGran_CFG.xml	*VIIRS Area Weight Mapping IP Granulation configuration guide
ProGipViirsWCalcGrid_CFG.xml	*VIIRS Area Weight Mapping IP Gridding configuration guide
ProSdrAtmsController_CFG.xml	*ATMS SDR Controller configuration guide
ProSdrAtmsRemapController_CFG.xml	*ATMS Remap SDR Controller configuration guide
ProSdrAtmsRemap_CFG.xml	*ATMS Remap SDR configuration guide
ProSdrAtmsVerifiedRDR_CFG.xml	*ATMS Verified RDR configuration guide
ProSdrAtms_CFG.xml	*ATMS SDR configuration guide
ProSdrCrisController_CFG.xml	*CrIS SDR Controller configuration guide
ProSdrCris_CFG.xml	*CrIS SDR configuration guide
ProSdrOmpsNpCal_CFG.xml	*OMPS Nadir Profile Calibration SDR configuration guide
ProSdrOmpsNpEarthController_CFG.xml	*OMPS Nadir Profile Earth SDR Controller configuration guide
ProSdrOmpsNpEarth_CFG.xml	*OMPS Nadir Profile Earth SDR configuration guide
ProSdrOmpsTcCal_CFG.xml	*OMPS Total Column Calibration SDR configuration guide
ProSdrOmpsTcEarthController_CFG.xml	*OMPS Total Column Earth SDR Controller configuration guide
ProSdrOmpsTcEarth_CFG.xml	*OMPS Total Column Earth SDR configuration guide
ProSdrViirsBrightPixel_CFG.xml	*VIIRS Bright Pixel SDR configuration guide
ProSdrViirsCal_CFG.xml	*VIIRS Calibration SDR configuration guide
ProSdrViirsController_CFG.xml	*VIIRS SDR Controller configuration guide
ProSdrViirsGeo_CFG.xml	*VIIRS Geolocation configuration guide
ProSdrViirsSolarDiffuser_CFG.xml	*VIIRS Solar Diffuser SDR configuration guide
ProSdrViirsVerifiedRdr_CFG.xml	*VIIRS Verified RDR configuration guide

* Indicates configuration guides only available after the IDPS Algorithms package has been extracted into the base ADL installation.

ADL uses an execution configuration file to allow a user to specify the items listed in the following table. These files are sometimes also referred to as “LW files” or “TK files” both in the documentation and software. This terminology has roots in the operational IDPS system. The items that are configurable in the execution configuration file are shown in the table below.

Table 3.2.4-2 Algorithm Execution Configurable Items

Description	Configurable Values
idpProcessName	The name of the executable being ran. This is used to name debug log files.
siSoftwareId	Present only to allow code reuse. Do not change for ADL.
isRestart	Present only to allow code reuse. Do not change for ADL.
useExtSiWriter	Present only to allow code reuse. Do not change for ADL.
debugLogLevel	Present only to allow code reuse. Do not change for ADL.
debugLevel	DBG_HIGH, DBG_MED, DBG_LOW, DBG_OFF. DBG_HIGH provides high level information while DBG_LOW provides very verbose low level messages.
dbgDest	D_FILE, D_CONSOLE, or D_BOTH to write debug output to a log file, the console, or both a log file and the console respectively.
dbgPath	The directory where log files should be written.
enablePerf	TRUE or FALSE to enable or disable performance timings
perfPath	The directory where performance output should be written.
domain	Present only to allow code reuse. Do not change for ADL.
subDomain	Present only to allow code reuse. Do not change for ADL.
startMode	Present only to allow code reuse. Do not change for ADL.
executionMode	Present only to allow code reuse. Do not change for ADL.
healthTimeoutPeriod	Present only to allow code reuse. Do not change for ADL.
lockinMem	Present only to allow code reuse. Do not change for ADL.
rootDir	The directory where request and messages log directories will be written. Note that this is different from debug logs.
inputPath	The directory where input data is located. Multiple paths can be separated by a colon. Additionally, environment variables can be used if surrounded by curly braces. For example: ${ADL_HOME}
outputPath	The directory where output data should be written.
dataStartIET	The IET time at which the data starts. This is only used when running without metadata.
dataEndIET	The IET time at which the data ends. This is only used when running without metadata.
actualScans	The actual number of scans for this data. This is only used when running without metadata.
previousActualScans	The actual number of scans for the previous granule of data (for cross granule inputs). This is only used when running without metadata.
nextActualScans	The actual number of scans for the next granule of data (for cross granule inputs). This is only used when running without metadata.
usingMetadata	TRUE or FALSE to run the algorithm in with metadata or no metadata mode respectively.
configGuideName	The name of the guide list file which lists all algorithm configuration files the algorithm needs to execute.
taskType	SDR, EDR, or SHUTDOWN
taskDetails1	Typically the granule ID
taskDetails2	Typically the granule version (A1)
taskDetails3	Typically the spacecraft name (NPP)
taskDetails4	Typically the sensor name (ATMS, CRIMSS, OMPS-NP, OMPS-TC, CRIS or VIIRS)

Algorithm execution configuration files are provided for the ADL compatible IDPS algorithms and are located in $ADL_HOME/cfg/dynamic/withMetadata and $ADL_HOME/cfg/dynamic/noMetadata. These algorithm execution configuration files are named according to the following naming convention:

> <executable name>LwFile.xml, where <executable name> is the name of an algorithm executable as listed in Table 3.1.2-1 “Software Inventory – Executables”.

3.2.5 Batch Files / Command Line Parameters

The ADL software does not use batch files. The only command line parameter to ADL processes is the execution configuration file.

3.2.6 Drives

The ADL software does not require any special drives be mapped. Input and Output (I/O) is done using the file system.

3.2.7 User/Service Accounts

Not Applicable to ADL.

4 INSTALLATION AND SETUP

4.1 Prerequisites

Prior to Installation of the ADL Software, the following are required

- HW/Operating System and Compiler Installation
- COTS Installation

If the ADL is deployed on an AIX or Linux platform, it is assumed that the platform administrator has sufficient knowledge to install the required ADL software, which is listed in section 3.2.

4.2 Installing the Software

This section describes the ADL 4.0 software and data as delivered by Raytheon. The installation and compilation steps assume you are working with the files contained on the media from the Raytheon delivery. For information and instructions pertaining to the ADL Virtual Appliance, created and maintained by the Space Science and Engineering Center (SSEC) at the University of Wisconsin-Madison, please reference the following website: https://jpss-adl-wiki.ssec.wisc.edu/mediawiki/index.php/ADL_Algorithm_Development_Library

ADL 4.0 is delivered as a collection of compressed archive files (gzipped tar files) spread across multiple dual-layered DVDs. We refer to these files as “packages” - there is an ADL base package, IDPS Algorithms package, and several packages of demonstration data. The ADL base package contains the ADL framework software, utility software, and template algorithm software - essentially all software except the IDPS algorithm software; it is a standalone compilable package. The IDPS algorithm software package contains all the IDPS ADL-compatible algorithm software and any necessary modified base package files; it is not a standalone compilable package and must be extracted into the base installation. The data packages contain demonstration input data (IDPS proxy test data) and truth output data for the algorithms; they also must be extracted into the base installation.

The ADL software is contained within adlSoftware.tgz, which is a complete stand-alone compilable package.

The IDPS algorithm software is contained within idpsAlgorithms.tgz, which must be built on top of the ADL software package. It cannot be compiled on its own.

The demonstration data has been packaged into different tar files. Due to some of the different sensors sharing data, the total package (of all DVDs) is larger than if it were contained within one large tar file. However, this method allows smaller subsets of the data to be downloaded as desired. Note that the data tar files will contain duplicate files, but if the data is extracted to the same ADL Installation directory, the duplicate files will simply overwrite the identical ones already on disk. Also, to help alleviate the strain on disk space, symbolic links

are used to point to the data files within each package and will be extracted with the specific Sensor DataLinks tar file.

The ATMS/CrIS/CrIMSS data is all contained within adlCrimssDataLinks.tgz.

The OMPS NP/OMPS TC data is all contained within adlOmpsDataLinks.tgz.

The VIIRS SDR algorithm data is contained within the following three tar files:

1. adlSdrViirsStaticInputDataLinks.tgz contains all the non-granule specific input data (Look Up Tables, coefficients files, etc) needed to run the VIIRS SDR algorithms.
2. adlSdrViirsDynamicInputDataLinks.tgz contains all the granule specific input data (RDRs, Tiles, etc) needed to run the VIIRS SDR algorithms for the example granules.
3. adlSdrViirsOutputDataLinks.tgz contains the output directories needed for running the VIIRS SDR algorithms and the truth output data for the VIIRS SDR example granules.

The VIIRS EDR algorithm data is described below. Due to its size, the dynamic input data was file was split into two tar files labeled Part1 and Part2. The VIIRS EDR algorithm data is contained within the following four tar files:

1. adlEdrViirsStaticInputDataLinks.tgz contains all the non-granule specific input data (Look Up Tables, coefficients files, etc) needed to run the VIIRS EDR algorithms.
2. adlEdrViirsDynamicInputDataLinks_Part1.tgz and adlEdrViirsDynamicInputDataLinks_Part2.tgz contain all the granule specific input data (SDRs, GEO files, etc) needed to run the VIIRS EDR algorithms for the example granules. These are separate tar files and SHOULD NOT be concatenated together. To extract these files, simply extract each tar file in the same location.
3. adlEdrViirsOutputDataLinks.tgz contains the output directories needed for running the algorithms and the truth output data for the VIIRS EDR example granules.

The ADL Algorithm Template data is all contained within the adlTemplateDataLinks.tgz file.

The HDF5 example files are all contained within adlHdfDataLinks.tgz.

The tiles required for global coverage of the sinusoidal Gridded IP tiles required to run the provided ADL algorithms are contained in adlGridIpDataLinks.tgz. Note that for tiles with the Collection Short Names of GridIP-VIIRS-Nbar-Ndvi-Rolling-Tile and GridIP-VIIRS-Snow-Ice-Cover-Rolling-Tile, data is provided that corresponds to Golden Day 1 and 2 (9/6/2002 and 1/25/2003, respectively). This data is not required for running the algorithms against the provided example granules.

The tiles required for global coverage of the polar stereographic tiles required to run the provided ADL algorithms are contained in adlTerEcoDataLinks.tgz. This data is not required for running the algorithms against the provided example granules.

Note that if the global tile data is used, the appropriate directories can be added to the <inputPath> section of the algorithm's *LwFile.xml. Multiple input paths can be used. Note that adding entire tile datasets to the input path significantly increase the time required to initialize and query the ADL data store.

Extraction Instructions:

1. Choose a local directory for the installation.
2. Download adlSoftware.tgz to the chosen directory.
3. Download idpsAlgorithms.tgz to the chosen directory, if desired.
4. Download all of the desired Sensor data packages (adl*DataLinks*.tgz), as describe above, into the chosen directory. If extracting VIIRS EDR data, you must get both Part1 and Part2 files.
5. From a shell, set into the directory containing all of the *.tgz files.
6. Extract the base ADL software package:
    a. /bin/gtar -zpxf adlSoftware.tgz
7. Extract the optional IDPS algorithm software package:
    a. /bin/gtar -zpxf idpsAlgorithms.tgz
8. Extract any or all of the desired adl*DataLinks.tgz files:
    a. /bin/gtar -zpxf adlTemplateDataLinks.tgz
    b. /bin/gtar -zpxf adlHdfDataLinks.tgz
    c. /bin/gtar -zpxf adlCrimssDataLinks.tgz
    d. /bin/gtar -zpxf adlOmpsDataLinks.tgz
    e. /bin/gtar -zpxf adlGridIpDataLinks.tgz
    f. /bin/gtar -zpxf adlTerEcoDataLinks.tgz
    g. /bin/gtar -zpxf adlMsdDataLinks.tgz

    NOTE: The adlGridIpDataLinks.tgz and adlTerEcoDataLinks.tgz data is not required for running the algorithms against the provided example granules.

9. If any VIIRS SDR algorithm is to be run with example data, execute the following three commands:
    a. /bin/gtar -zpxf adlSdrViirsStaticInputDataLinks.tgz
    b. /bin/gtar -zpxf adlSdrViirsDynamicInputDataLinks.tgz
    c. /bin/gtar -zpxf adlSdrViirsOutputDataLinks.tgz

    If the desired VIIRS SDR algorithms are to be run only using data for granules other than those provided in the example data, you only need to execute the following command:

    /bin/gtar -zpxf adlSdrViirsStaticInputDataLinks.tgz

    NOTE: you will also have to update the *LwFile.xml and create output directories. See the ADL User's Manual Part 2, section 5.2.3 for more information.

10. If any VIIRS EDR algorithm is to be run with example data, execute the following three commands:

a. /bin/gtar -zpxf adlEdrViirsStaticInputDataLinks.tgz
b. /bin/gtar -zpxf adlEdrViirsDynamicInputDataLinks_Part1.tgz
c. /bin/gtar -zpxf adlEdrViirsDynamicInputDataLinks_Part2.tgz
d. /bin/gtar -zpxf adlEdrViirsOutputDataLinks.tgz

If the desired VIIRS EDR algorithms are to be run only using data for granules other than those provided example data, you only need to execute the following command:

/bin/gtar -zpxf adlEdrViirsStaticInputDataLinks.tgz

NOTE: you will also have to update the *LwFile.xml and create output directories. See the ADL User's Manual Part 2, section 5.2.3 for more information.

11. After performing the required Step 6 and any of the optional steps, a folder called ADL will be extracted into the current directory. This folder contains all of the software for the ADL framework, plus any of the IDPS algorithm software and demonstration data extracted from the packages in the optional steps. If running an algorithm is attempted when its data was not extracted, the following error messages will be seen in the log file for the algorithm (for example):

    ROOT PRO_FAIL Could not create DMS client: Invalid instance path from file ProCmnAppl.cpp

    The DMS directory is not valid:

    /testdata/ADL_3.1/ADL/data/input/noMetadata/ProSdrOmpsNpEarthInputs

    This indicates that the example data for that algorithm was not downloaded and extracted. This will also occur if new data was created, but the *LwFile.xml was not updated to point to the new directory.

12. Remove write permission on everything in the data repository directory to safeguard the input test data from accidentally being deleted, by running the following command:
    a. chmod -R a-w ADL/data/repositories
13. Optionally, download the correspondingly named adl*.md5 files into the chosen directory by running the following commands:
    a. cp *.md5 <full path of chosen directory>
    b. md5sum -c PACKAGE_NAME.md5

    Where PACKAGE_NAME is the name of the adl*Software or adl*DataLinks tar files that were downloaded. This will verify the MD5 checksum of all files in each of the tar files. All checksums should match.

NOTES:

1. The gtar commands may need to be modified for the name of the GNU tar command (tar or gtar) on your system.

2. The compressed tar file was created with GNU tar version 1.19.

Compilation Instructions:

1. From a shell, set into the directory chosen in Step 1 of the "Extraction Instructions", then change to the ADL directory.
2. One time only, make writeable the envSetup.* files in the build subdirectory. Edit these files to specify the locations of the COTS products on your machine(s). The environment variables listed in section 3.2.2 must be updated in envSetup.ksh or envSetup.csh The "if-then" statements for each host can be removed to leave a single set of environment variable setups, if desired. To do this, you should use the '$HOST == "auislnx01x"' or '$OSTYPE == "AIX"' examples for the variables required.

   Edit the ADL_HOME setting to specify the directory chosen in Step 1, making sure to put the /ADL at the end of the chosen directory name.

   Both the envSetup.csh and envSetup.ksh scripts should be modified accordingly. If significant changes were made to envSetup.* (such as new variables or library name changes), you may need to also update the Environment checker script located in script/checkEnv.pl.

3. Set the required environment variables by sourcing the appropriate envSetup.* script.
   a. source build/envSetup.csh OR
   b. source build/envSetup.ksh
4. The Imake parameters listed below must be updated in either the XL or GCC section of $ADL_HOME/imakeconf/IDPS_Conf_Params. The use of a full file path for each parameter is recommended.
   a. JC_Cmd – Java compiler
   b. CC_Cmd – C++ compiler
   c. CCr Cmd – Thread safe C++ compiler
   d. FC_Cmd – Fortran compiler
   e. FCr_Cmd – Thread safe Fortran compiler
5. The XML Editor parameters listed below must be updated in $ADL_HOME/java/pro/cat3/xml/support/XmlEditor.properties. Use of a full file path for each parameter is required. Additionally, the file paths must be prefixed with “file:”. Note: If this properties file is ever updated again after the initial compile, you need to rebuild the xml editor for the changes to take effect.
   a. schema.pro – PRO schema file
   b. schema.cdfcb – CDFCB scheme file
   c. config.log4j – log4j configuration file
6. Compile the ADL software. After environment variables, imakeconf parameters and XML Editor parameters are updated, the software should then be compiled. There are several ways to do this and several scripts provided which utilize the ADL_HOME variable to find the source code location. The

three options are: a manual process, a single threaded shell script, and a multi-threaded perl script. Each of these options assumes that Step 3 has been completed. Note that each of the scripts used below has other possible options for different modes of compilation, such as diagnostics mode. Execute the scripts with the -h option to see a list of available options.

a. Manual process:

```
# set to the home directory
cd $ADL_HOME
# Generate the top-level Makefile with Debug turned on:
build/make_one_imake.csh -d
# Build all of the ADL software:
make imake clean src library program
```

b. Single threaded shell script:

```
# set to the home directory
cd $ADL_HOME
# Run the single threaded build script which always turns Debug on.
# Run the appropriate script for your selected shell:
./buildAdl.csh      OR
./buildAdl.ksh
```

c. Multi-threaded perl script:

```
# set to the home directory
cd $ADL_HOME
# Run the multi-threaded build script with Debug turned on.
# Debug can be turned off and the code Optimized by leaving off
# the "-debug" option.  This command uses "tee" to pipe the
# high-level script output into the "build.out" file whereas
# the detailed compile results will be stored in "build.log".
# This script spawns off multiple processes in parallel in an
# attempt to compile as quickly as possible.  Note that on some
# machines which cannot handle the multiple threads well,
# this script may have problems compiling.
# To remedy this, you can simply rerun the script with only
# the -library and/or -program options to separate the steps.
# Enter the following command (on one line):
script/build_adl.pl -clean -makefiles -debug -log build.log -src
 -library -program -env | tee build.out
```

The ADL software from the provided DVDs should now be installed into an appropriate location and fully built.

To verify that the software compiled successfully, ensure the executables found in Table 3.1.2-1 Software Inventory – Algorithm Executables exist in the $ADL_HOME/bin directory, and the XML Editor / Binary Editor jar files exist in the $ADL_HOME/java/dist directory at the end of compilation.

Execute the ADL provided demonstration algorithms to validate proper install.

- runDemoAlgorithms.ksh

Verify that the demonstration algorithms run successfully and produce output products.

4.3 Configuring the Software

Not Applicable.

5 MISCELLANEOUS NOTES

5.1 Possible Compiler Warnings

Depending on the platform used to compile the ADL software, several warnings may be expected. All of these warnings are hidden or suppressed in the operational code, and will also be hidden or suppressed when compiling the ADL software on the AIX platform.

These warnings will show up in the ADL code on the Linux platforms because the GCC compiler is used with the –Wall flag (all warnings) turned on. The majority of the warnings seen will be in regards to the C++ compiler, but there are several warnings produced by the Fortran science code or by options not supported by different versions of the gfortran compiler. The vast majority of the warnings originate from the CAT1 code, but updating the CAT1 code to remove warnings was outside the scope of ADL Phase 4 development. The warnings can be safely ignored, however, they do point out items that should be corrected in the future.

Below is a list of compiler warnings typically seen with the gcc version 4.4.0 20090514 (Red Hat 4.4.0-6) compiler when building ADL. It's important to note that if a different compiler is used or if different options are provided, new or different warnings may be seen. Note that this is not an exhaustive list, these warnings are only an example, and in many cases the format and explanation provided are specific to the gcc/gfortran 4.4 compiler.

Fortran warnings:

- Warning: Deleted feature: Start expression in DO loop at (1) must be integer
  - The OMPS Science code is using an array defined as a REAL in the index in an FOR-LOOP where the rest of indexes are integers.
- Warning: Deleted feature: End expression in DO loop at (1) must be integer
  - Similar to the message above, the OMPS Science code is using an array defined as a REAL in the index in an FOR-LOOP where the rest of indexes are integers.
- Warning: Deleted feature: GOTO at (1) jumps to END of construct at (2)
  - The OMPS Science code uses a Label on the END of a DO-LOOP.
- Warning: Label at (1) is not in the same block as the GOTO statement at (2)
  - The OMPS Science code has a GOTO in one IF-Block that jumps into another IF-Block. The code is correct, but the compiler is simply warning of the danger in doing this type of GOTO statement.
- Warning: Using -M <directory> is deprecated, use -J instead.
  - This warning occurs because multiple Linux environments are used in testing ADL. The –M option specifies the location to store the .mod files created by gfortran. In the 4.3 version only the –M option exists, but in the 4.4 version the –J option is preferred. This warning is simply informing the user that the –M option may be going away in future releases.

C++ Warnings:

1) <directory>/<file>: warning: â M-^@M-^Xtypedefâ M-^@M-^X was ignored in this declaration
   a) This is a C typedef of a struct, union, or enum declaration, but these typedefs are not necessary in C++. This warning is a case where the GCC compiler is being more strict than the AIX compiler. Such typedefs are harmless, so this warning should be ignored.
2) <directory>/<file>: warning: deprecated conversion from string constant to âchar*â
   a) This is an implicit conversion from a string to a char pointer. This is an example of a warning ignored in the AIX compiler, but produced by GCC due to adherence to strict typecasting rules.
3) <directory>/<file>: warning: â<variable OR type>â defined but not used
   a) This is a warning produced by GCC because the specified type name is not used in the file where it is defined. The overwhelming majority of these warnings should be ignored because the specified type name is used in a file which includes the file where the warning is reported.
4) <directory>/<file>: warning: unused variable â<variable name>â
   a) This is a warning produced by GCC because the specified variable is not used in the file where it is defined. These warnings are harmless, and most are accurate because the variable is not actually used. However, since many are in Science code, the code has not been updated to remove them. The code could be updated, but this task is outside of the scope of ADL.
5) <directory>/<file>: warning: comparison between signed and unsigned integer expressions
   a) This warning is due to more strict typecasting rules of GCC. This warning is another case where the GCC compiler is being more strict than the AIX compiler.
6) <directory>/<file>: warning: non-local variable â<anonymous enum> <type_name>â uses anonymous type
   a) This warning is similar to warning #1 above. This is a case where GCC is complaining because the name of the enum was placed after the enumeration list instead of before it. This sort of warning is also harmless.
7) <directory>/<file>: warning: missing braces around initializer for âM-^@M-^X<type_name>âM-^@M-^Y
   <directory>/<file>: warning: suggest parentheses around comparison in operand of â<operator>â
   <directory>/<file>: warning: multi-line comment
   a) These are all more examples of more strict checking done by GCC where it is complaining that a C-type syntax has been used instead of a C++ type syntax.
8) <directory>/<header file>: In constructor âM-^@M-^X<constructor_definition>âM-^@M-^Y:
   <directory>/<header file>:<line>: warning: âM-^@M-^X<member_variable_1>âM-^@M-^Y will be initialized after
   <directory>/<header file>:<line>: warning: âM-^@M-^X <member_variable_2>âM-^@M-^Y
   <C++ file>:<line>: warning: when initialized here
   a) This warning is notifying the programmer that the order of the member variables in the constructor initialization does not match the order that they are defined in the header file. In some cases this may be important and could lead to incorrectly initialized values.

In most cases, this is warning about integer initialization that is not dependent upon the other member variables and is therefore harmless.